

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION  
International Bureau



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>6</sup> : <b>H04N 7/50, 7/36</b>	<b>A1</b>	(11) International Publication Number: <b>WO 99/16252</b> (43) International Publication Date: 1 April 1999 (01.04.99)
-----------------------------------------------------------------------------------	-----------	---------------------------------------------------------------------------------------------------------------------------

(21) International Application Number: PCT/US98/19097

(22) International Filing Date: 14 September 1998 (14.09.98)

(30) Priority Data:

60/059,678	19 September 1997 (19.09.97)	US
09/001,129	30 December 1997 (30.12.97)	US
09/001,122	30 December 1997 (30.12.97)	US

(71) Applicant: SONY ELECTRONICS INC. [US/US]; 1 Sony Drive, Park Ridge, NJ 07656 (US).

(72) Inventors: BOSE, Subroto; 3516 Oxford Drive, Santa Clara, CA 95051 (US). GADRE, Shirish, C.; 1265 N. Capitol Avenue #78, San Jose, CA 94132 (US). OZCELIK, Taner; 542 Military Way, Palo Alto, CA 94306 (US). PALUCH, Edward, J.; 3660 Vireo Avenue, Santa Clara, CA 95051 (US). REZA, Syed; 870 Fremont Street #105, Santa Clara, CA 95050-4863 (US).

(74) Agents: JORDAN, Joseph, R. et al.; Wood, Herron & Evans, L.L.P., 2700 Carew Tower, Cincinnati, OH 45202 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

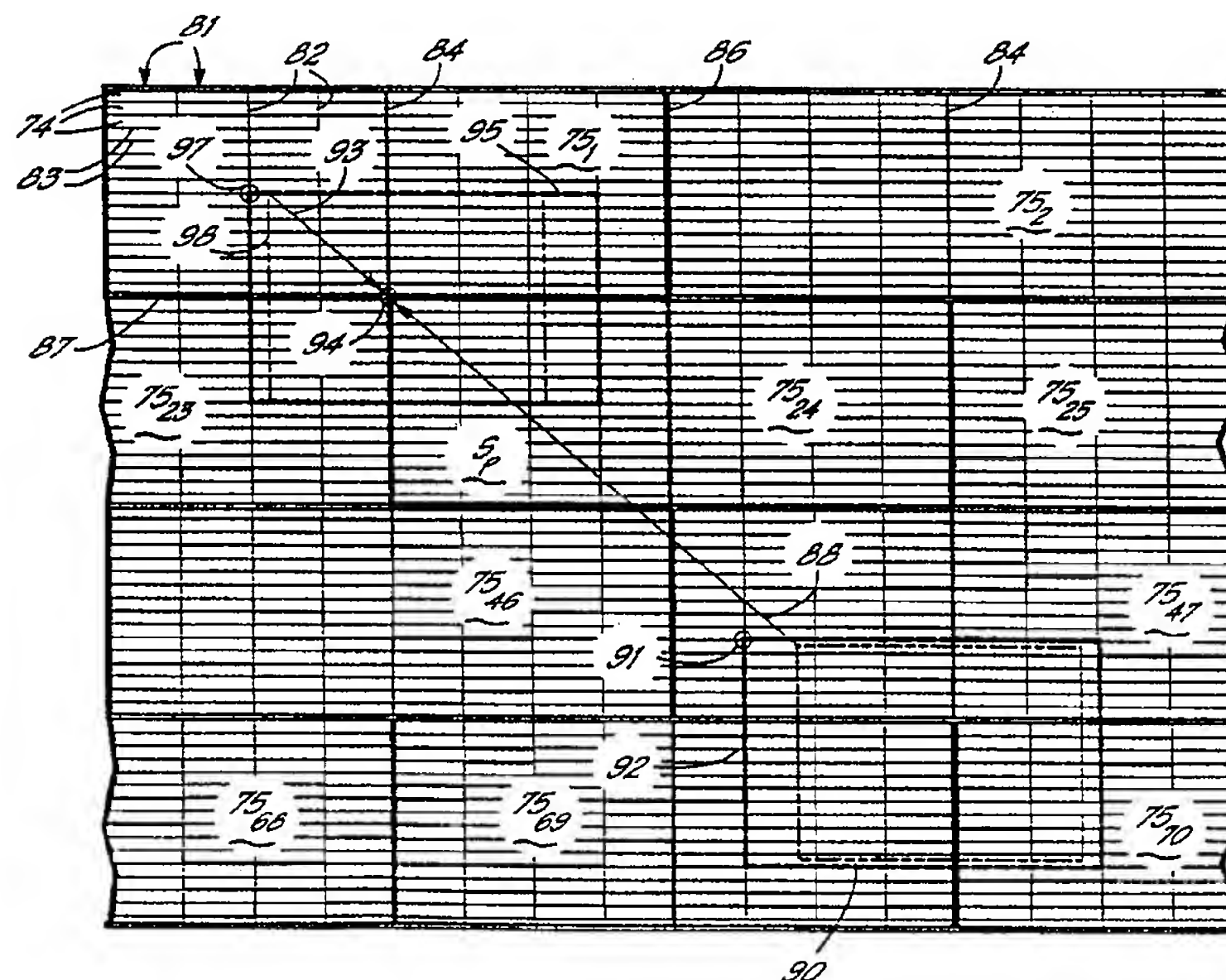
*With international search report.*

(54) Title: MOTION COMPENSATED DIGITAL VIDEO DECODING WITH BUFFERED PICTURE STORAGE MEMORY MAP

(57) Abstract

A digital video presentation system (30) is provided with hardware and software logic for mapping the picture data into buffer memory (48) in a way that permits both the reading of motion vector compensated macroblocks (S) of data and the reading of horizontal picture wide scan lines with a low number of memory page crossings. Preferably, the memory (48) is a plurality of rows (74), for example 16 rows, wide. Preferably, 16 lines of 8-pixel (two 32 pixel wide column) line segments of 8x8 pixel blocks (A-P) are stored in consecutive storage locations (76) followed by the consecutive storage vertically adjacent line segments until one line segment is stored in each logical row (74) of the memory (48). Then the next horizontally adjacent set of line segments of similarly stored until the right boundary of the picture is reached, then the each additional row of 16 lines of the picture similarly are stored until the bottom of the picture is reached. Each 16x16 pixel macroblock (S) of data is stored on a single page (75); preferably, two horizontally adjacent macroblocks (S) are stored on one page of memory (48). Each line of the picture is stored in contiguous locations on the same row of the memory.

The motion compensation logic (77) interprets motion vectors from the incoming data (32) and calculates addresses for a macroblock (S) of picture data by separating read commands into separate commands where a page boundary divides the macroblock into vertically adjacent rectangles. Memory controller logic (50) further divides such rectangles where they cross boundaries between horizontally adjacent pages of the memory. One fixed address 8-hex increment steps from line segment to vertically adjacent line segment while another fixed address 80-hex increment steps horizontally from one 8-pixel line segment to the next, such as across a scan line of the picture.



***FOR THE PURPOSES OF INFORMATION ONLY***

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

<b>AL</b>	Albania	<b>ES</b>	Spain	<b>LS</b>	Lesotho	<b>SI</b>	Slovenia
<b>AM</b>	Armenia	<b>FI</b>	Finland	<b>LT</b>	Lithuania	<b>SK</b>	Slovakia
<b>AT</b>	Austria	<b>FR</b>	France	<b>LU</b>	Luxembourg	<b>SN</b>	Senegal
<b>AU</b>	Australia	<b>GA</b>	Gabon	<b>LV</b>	Latvia	<b>SZ</b>	Swaziland
<b>AZ</b>	Azerbaijan	<b>GB</b>	United Kingdom	<b>MC</b>	Monaco	<b>TD</b>	Chad
<b>BA</b>	Bosnia and Herzegovina	<b>GE</b>	Georgia	<b>MD</b>	Republic of Moldova	<b>TG</b>	Togo
<b>BB</b>	Barbados	<b>GH</b>	Ghana	<b>MG</b>	Madagascar	<b>TJ</b>	Tajikistan
<b>BE</b>	Belgium	<b>GN</b>	Guinea	<b>MK</b>	The former Yugoslav Republic of Macedonia	<b>TM</b>	Turkmenistan
<b>BF</b>	Burkina Faso	<b>GR</b>	Greece			<b>TR</b>	Turkey
<b>BG</b>	Bulgaria	<b>HU</b>	Hungary	<b>ML</b>	Mali	<b>TT</b>	Trinidad and Tobago
<b>BJ</b>	Benin	<b>IE</b>	Ireland	<b>MN</b>	Mongolia	<b>UA</b>	Ukraine
<b>BR</b>	Brazil	<b>IL</b>	Israel	<b>MR</b>	Mauritania	<b>UG</b>	Uganda
<b>BY</b>	Belarus	<b>IS</b>	Iceland	<b>MW</b>	Malawi	<b>US</b>	United States of America
<b>CA</b>	Canada	<b>IT</b>	Italy	<b>MX</b>	Mexico	<b>UZ</b>	Uzbekistan
<b>CF</b>	Central African Republic	<b>JP</b>	Japan	<b>NE</b>	Niger	<b>VN</b>	Viet Nam
<b>CG</b>	Congo	<b>KE</b>	Kenya	<b>NL</b>	Netherlands	<b>YU</b>	Yugoslavia
<b>CH</b>	Switzerland	<b>KG</b>	Kyrgyzstan	<b>NO</b>	Norway	<b>ZW</b>	Zimbabwe
<b>CI</b>	Côte d'Ivoire	<b>KP</b>	Democratic People's Republic of Korea	<b>NZ</b>	New Zealand		
<b>CM</b>	Cameroon	<b>KR</b>	Republic of Korea	<b>PL</b>	Poland		
<b>CN</b>	China	<b>KZ</b>	Kazakstan	<b>PT</b>	Portugal		
<b>CU</b>	Cuba	<b>LC</b>	Saint Lucia	<b>RO</b>	Romania		
<b>CZ</b>	Czech Republic	<b>LI</b>	Liechtenstein	<b>RU</b>	Russian Federation		
<b>DE</b>	Germany	<b>LK</b>	Sri Lanka	<b>SD</b>	Sudan		
<b>DK</b>	Denmark	<b>LR</b>	Liberia	<b>SE</b>	Sweden		
<b>EE</b>	Estonia			<b>SG</b>	Singapore		

- 1 -

**MOTION COMPENSATED DIGITAL VIDEO DECODING  
WITH BUFFERED PICTURE STORAGE MEMORY MAP**

The present invention relates to digital video decoding, and more particularly, to buffer memory configuration, buffer memory address generation and video decoder implementation for the reproduction of moving pictures from digital signals, such as in broadcast television receiver systems, digital video compact disc playing systems and other digital video presentation systems.

5     **Background of the Invention**

Recent times have seen an acceleration in efforts by suppliers of consumer electronics to greatly expand the amount and quality of information provided to users. The expanded use of multimedia information in communications and entertainment systems along with user demands for higher quality and faster presentations of the information has driven the communications and entertainment industries to seek systems for communicating and presenting information with higher densities of useful information. These demands have stimulated the development and expansion of digital techniques to code and format signals to carry the information.

10     Unlike most of the communication systems of the past, particularly television broadcast systems and other systems used for home entertainment, where analog signals have filled available bandwidths with single program real time signals in a straight forward format that includes much redundant information as well as much humanly imperceivable information, digital transmission systems possess the ability to combine and identify multiple programs and to selectively filter out redundant or otherwise useless information to provide capabilities for the transmission of programs having higher quality or having higher useful information carrying ability or density. As a result of the high technological demand for such capabilities, advances toward the specification and development of digital communications formats and systems have accelerated.

15     In furtherance of these advances, the industry sponsored Motion Pictures Expert Group (MPEG) chartered by the International Organization for Standardization (ISO) has specified a format for digital video and two channel stereo audio signals that has come to be known as MPEG-1, and, more formally, as ISO-11172. MPEG-1 specifies formats for representing data inputs to digital decoders, or the syntax for data bitstreams that will carry programs in digital formats that decoders can reliably decode. In practice, the MPEG-1 standards have been used for recorded programs that are usually read by software systems. The program signals include digital data of various programs or program components with their digitized data streams multiplexed together by

parsing them in the time domain into the program bitstreams. The programs include audio and video frames of data and other information.

An enhanced standard, known colloquially as MPEG-2 and more formally as ISO-13818, has more recently been agreed upon by the ISO MPEG. This enhanced standard has grown out of needs for specifying data formats for broadcast and other higher noise applications, such as high definition television (HDTV), where the programs are more likely to be transmitted than recorded and more likely to be decoded by hardware than by software.

The MPEG standards define structure for multiplexing and synchronizing coded digital and audio data, for decoding, for example, by digital television receivers and for random access play of recorded programs. The defined structure provides syntax for the parsing and synchronizing of the multiplexed stream in such applications and for identifying, decoding and timing the information in the bitstreams.

The MPEG video standard specifies a bitstream syntax designed to improve information density and coding efficiency by methods that remove spacial and temporal redundancies. For example, the transformation blocks of 8 x 8 luminance pels (pixels) and corresponding chrominance data using Discrete Cosine Transform (DCT) coding is contemplated to remove spacial redundancies, while motion compensated prediction is contemplated to remove temporal redundancies. For video, MPEG contemplates Intra (I) frames, Predictive (P) frames and Bidirectionally Predictive (B) frames. The I-frames are independently coded and are the least efficiently coded of the three frame types. P-frames are coded more efficiently than are I-frames and are coded relative to the previously coded I- or P frame. B-frames are coded the most efficiently of the three frame types and are coded relative to both the previous and the next I- or P-frames. The coding order of the frames in an MPEG program is not necessarily the same as the presentation order of the frames. Headers in the bitstream provide information to be used by decoders to properly decode the time and sequence of the frames for the presentation of a moving picture.

The video bitstreams in MPEG systems include a Video Sequence Header containing picture size and aspect ratio data, bit rate limits and other global parameters. Following the Video Sequence Header are coded groups-of-pictures (GOPs). Each GOP usually includes only one I-picture and a variable number of P- and B-pictures. Each GOP also includes a GOP header that contains presentation delay requirements and other data relevant to the entire GOP. Each picture in the GOP includes a picture header that contains picture type and display order data and other information relevant to the picture within the picture group.

Each MPEG picture is divided into a plurality of Macroblocks (MBs), not all of which need be transmitted. Each MB is made up of 16 x 16 luminance pels, or a 2 x 2 array of four 8 x 8 transformed blocks of pels. MBs are coded in Slices of consecutive variable length strings of MBs, running left to right across a picture. Slices may begin and end at any intermediate MB position of the picture but must respectively begin or end whenever a left or right margin of the picture is encountered. Each Slice begins with a Slice Header that contains information of the vertical position of the Slice within the picture, information of the quantization scale of the Slice and other information such as that which can be used for fast-forward, fast reverse, resynchronization in the event of transmission error, or other picture presentation purposes.

The Macroblock is the basic unit used for MPEG motion compensation. Each MB contains an MB header, which, for the first MB of a Slice, contains information of the MB's horizontal position relative to the



left edge of the picture, and which, for subsequently transmitted MBs of a Slice, contains an address increment. Not all of the consecutive MBs of a Slice are transmitted with the Slice.

The presentation of MPEG video involves the display of video frames at a rate of, for example, twenty-five or thirty frames per second (depending on the national standard used, PAL or NTSC, for example). Thirty frames per second corresponds to presentation time intervals of approximately 32 milliseconds. The capacity of MPEG signals to carry the information necessary for HDTV and other presentations providing high resolution video is achieved in part by exploiting the concept that there is typically a high degree of correlation between adjacent pictures and by exploiting temporal redundancies in the coding of the signals. Where two consecutive video frames of a program are nearly identical, for example, the communication of the consecutive frames requires, for example, only the transmission of one I-picture along with the transmission of a P-picture containing only the information that differs from the I-picture, or Reference Picture, along with information needed by the decoder at the receiver to reconstruct the P-picture from the previous I-picture. This means that the decoder must have provision for storage of the Reference Picture data.

Information contained in a P-picture transmission includes blocks of video data not contained in a Reference I- or P-picture as well as data needed to relocate in the picture any information that is contained in the previous I- or P-picture that has moved. The technique used in MPEG systems to accomplish P-picture construction from a Reference picture is the technique of Forward Prediction in which a Prediction Error in the form of a Prediction Motion Vector (MV) is transmitted in lieu of the video data of a given or Target MB. The MV tells the decoder which MB of the I- or P- Reference Picture, is to be reproduced as the Target MB.

With B-pictures, a Bidirectional Temporal Prediction technique called Motion Compensated Interpolation, is used. Motion Compensated Interpolation is accomplished by transmitting, in lieu of the video data for a Target MB, an MV that specifies which MB to copy either from the previous Reference Picture or from the next future Reference Picture, or from the average of one MB from each of the previous and next future Reference Pictures.

An MPEG Motion Compensated Prediction video decoder of a type that is practical for HDTV must possess a Reference Picture data storage capability that permits construction by the receiver decoder of the B- and P-frames containing the motion vectors that specify MB of the reference pictures. In order to provide sufficient data retrieval speed to perform the motion compensation calculations, static or on chip memory (SRAM) could be provided. The use of SRAM having the storage capacity necessary to store a video picture is a straight-forward but expensive way to provide the capability. The use of an off-chip DRAM buffer as an alternative to on-chip static ram, however, presents the problem of memory access time exceeding the interframe time availabilities of the program. For example, the specification in the MVs of the MBs that must be retrieved to perform the motion compensation prediction calls for access to the storage medium in an order that has a substantial random component. DRAM memory is by nature divided into memory segments called "pages". Consecutive reads of the memory within a page require substantially less time than consecutive reads that call for the crossing of page boundaries. The random memory access requirements of motion compensation prediction results in a number of memory page crossings, which can result in memory access time requirements that prevent the efficient and effective use of DRAM memory buffers.

In addition to the retrieval of MBs from reference pictures, the storage and retrieval of data of individual pels can adversely affect the efficiency of the decoding process. This is particularly the case with post filtering, where low pass filtering is applied pixel-to-pixel to remove rapid spacial fluctuations in values. For the presentation of video, the decoder must generate both a luminance (overall brightness or intensity) value for each pel and a chrominance (color pair) value for each pel. The MPEG-2 standard, however, calls for the ability to decode color video programs at bit rates as low as 4 Mbits per second, as for both progressive (non-interleaved) or interleaved video.

With interleaved video, a video frame is formed of two fields, one containing the even scan lines of a picture (the "top field") and one containing the odd scan lines of the picture (the "bottom field"). The fields are alternately output to a video display in each 32 millisecond cycle, allowing 16 milliseconds for each field to be output. Certain standards such as the CCIR-601 standard, which must be supported by MPEG, include an interleaved format. For interleaved video motion compensation in MPEG-1, all pictures are frame pictures that include both the top field and the bottom field, but in MPEG-2, the I-frames, P-frames and B-frames may be either full video frames of both top and bottom fields or may include only a top field or a bottom field.

Further, depending on the bitrate and format employed, one chrominance pair may be coded for each luminance value. This is referred to as a 4:4:4 chrominance format, and requires the highest bitrate or coding efficiency. Alternatively, other formats provide for one chrominance value for each two or four luminance values, by subsampling chrominance 2:1 horizontally, or both horizontally and vertically. Such formats are referred to as the 4:2:2 format and the 4:2:0 format, respectively. With interleaved pictures, where a picture of alternating top rows of luminance pels is first transmitted and then a picture of alternating bottom rows is transmitted, alternating rows of chrominance pair pel values are transmitted with fields of the luminance pels, with those chrominance pels transmitted with the "top" field relating to a 2 x 2 array of top field luminance values, while those chrominance pels transmitted with the "bottom" field relate to an interleaved 2 x 2 array of bottom field luminance values. As a result, a straight-forward storage and retrieval of luminance and chrominance data in the order it is received can complicate and substantially slow the decoding process.

In particular, MPEG-2 video decoders must decode signals with interleaved video in what has been called, and referred to above as, the CCIR-601 (and which has also been called the ITU-R) color video format, where each pixel is coded as a luminance 8 bit value sampled at a 13.5 MHz rate along with a red chrominance value and a blue chrominance value, 8 bits each and sampled at a 6.75 MHz rate. In this format, the video frames are 720 pels per line, and either 480 lines per frame at 30 frames per second or 576 lines per frame at 25 frames per second. Uncompressed, this requires 216 Mbits/s, but the signal may be compressed to as few as 2 Mbits/s, with 4 Mbits/s being a typical rate.

Each of the formats referred to above and other formats, together with the variety of formats that MPEG receivers must decode, make it difficult to effectively and efficiently buffer the data for the video being reproduced at the receiver. Accordingly, in the decoding and reproduction of MPEG video programs, there is a need for an effective and efficient memory usage scheme, particularly for performing Motion Compensation Prediction and post filtering.

**Summary of the Invention**

A primary objective of the present invention is to provide for the efficient storage of digital video information at the decoder of digital receiver, and particularly, to provide for the time and memory efficient storage and retrieval of digital video information in a receiver such as of an MPEG-2 video presentation system.

5 It is a further particular objective of the present invention to provide a memory buffer memory preferably of a DRAM type and a storage order of video data therein that minimizes the frequency of page crossings during retrieval of the data.

10 In accordance with the principles of the present invention, there is provided a buffer memory storage sequence by which video data is stored by macroblock in buffer memory with a plurality of macroblocks stored on the same buffer memory page. In particular, the data of a video picture are mapped into the buffer memory in an order that exploits the likelihood that consecutive reads and writes of the memory will involve macroblocks stored on the same memory page, thereby reducing page crossings during memory access and increasing data storage and retrieval speed. In the preferred embodiment of the invention, the memory mapping utilizes a recurring pattern that extends the advantages of the invention to buffer memories of differing page sizes.

15 Preferably, data is stored with scan lines within a macroblock and between adjacent macroblocks having an increased incidence of lying in the same line of a memory array, and thus stored consecutively in the memory and on the same page of the buffer memory. In accordance with such principles of the present invention, more than one horizontally adjacent macroblock stored consecutively in memory, preferably with the same scan lines of both stored consecutively, preferably on the same DRAM page, and preferably with scan line data stored at constant increments.

20

In accordance with the preferred embodiment of the present invention, the data representing each of the four 8 x 8 pixel video blocks of each macroblock are stored, line by line, in the sequence of (A) upper left block, (C) lower left block, (B) upper right block and (D) lower right block. This pattern is repeated for each macroblock, from the left to the right edge of the picture, of each horizontal row of macroblocks, from the top to the bottom of the picture.

25

Preferably, the buffer memory includes three sections each capable of storing the video data for one picture. One section stores the data of one I- or P-frame, one section stores the data of the next subsequent P-frame, while the third section stores the reconstructed or interpolated picture as the result of a received B-frame. The buffer memory sections are each divided into two halves, one for storing *Y* or luminance data for the respective picture and one for storing *U, V* or chrominance data for the respective picture. Also preferably, the I and P frame pictures are full frame pictures that include alternating rows of top and bottom field data interleaved as the data appears in a display, while the B-frame pictures are preferably for one or the other field only, with the corresponding section of the memory for the B-frame pictures being divided into two parts each capable of storing data for one field.

30

35 In accordance with the preferred embodiment of the invention, the buffer memory is logically arranged in an array of storage positions 16 bits or rows wide and divided into 32 byte wide columns, or columns wide enough to store video data for eight 32 bit pixels, or one line of a video block. Each column will thus store picture data in the formats discussed above with one block per column of a interleaved (both even and odd line

for I- and P-frame) video picture or two blocks per column of a field (even or odd lines only for B-frame) of a video picture.

The memory mapping and data storage concepts of the preferred embodiment, each page of, for example, 4096 bytes, will store two horizontally adjacent macroblocks of interleaved video. Alternatively, such a page will store a four horizontally adjacent macroblocks of any one even or odd line field of non-interleaved video. As a result, either two or four horizontally adjacent macroblocks are stored on the same line of memory of the same memory page for convenient outputting to post-filtering hardware. Where larger memory pages are used, more horizontally adjacent macroblocks will be stored on the same page of memory.

Further, when a 16x16 or 17x17 pixel square of data is retrieved from or written to buffer memory, the number of pages of memory that must be addressed is minimized. This provides efficiency when copying data from an I-frame or a P-frame when reconstructing a P-frame or a B-frame picture.

These and other objectives and advantages of the present invention will be more readily apparent from the following detailed description of the preferred embodiments of the invention, in which:

**Brief Description of the Drawings:**

Fig. 1 is a block diagram representing an MPEG-2 receiver embodying principles of the present invention.

Fig. 2 is a block diagram representing the ASIC portion of the receiver of Fig. 1.

Figs. 3 and 3A are diagrammatic representations of the upper left corner of a non-interleaved television picture of CCIR-601 4:2:0 and 4:2:2 chrominance format, respectively, with pel luminance and chrominance data variables noted thereon.

Fig. 4 is a diagrammatic representation of the upper left corner of an interleaved television picture of CCIR-601 4:2:0 chrominance format with pel luminance and chrominance data variables noted thereon.

Fig. 5 is a diagrammatic representation of the upper left corner of an interleaved television picture of CCIR-601 4:2:2 chrominance format with pel luminance and chrominance data variables noted thereon.

Figs. 6 diagrammatically represents the pel block arrangement of a television picture.

Fig. 7 is a detailed block diagram representing the DRAM buffer and video decoder portions of the MPEG-2 receiver of Figs 1 and 2.

Fig. 7A is a detailed diagram representing the motion compensation section of the video decoder portion of the receiver of Fig. 7.

Fig. 8 is a buffer memory map diagram of stored luminance data for full frame I and P pictures in accordance with one preferred embodiment of the present invention. Fig. 8A is a buffer memory map similar to Fig. 8 of stored luminance data for one field B pictures in accordance with one preferred embodiment of the present invention.

Fig. 9 is a buffer memory map diagram similar to Fig. 8 of stored chrominance data for full frame I and P pictures of 4:2:2 chrominance format in accordance with one preferred embodiment of the present invention.

Fig. 9A is a buffer memory map diagram similar to Fig. 9 of stored chrominance data for one field B pictures of 4:2:2 chrominance format in accordance with one preferred embodiment of the present invention.



Fig. 10 is a buffer memory map diagram similar to Fig. 9 of stored chrominance data for full frame *I* and *P* pictures of 4:2:0 chrominance format in accordance with one preferred embodiment of the present invention.

Fig. 10A is a buffer memory map diagram similar to Fig. 10 of stored chrominance data for one field *B* pictures of 4:2:0 chrominance format in accordance with one preferred embodiment of the present invention.

Fig. 11 is a larger scale diagram of the buffer memory map diagrams of Figs. 8 and 9.

Fig. 11A is a diagram similar to Fig. 11 of the buffer memory map diagrams of Figs. 8A and 9A.

Fig. 11B is a diagram similar to Fig. 11A of the buffer memory map diagram of Fig. 10A.

Fig. 12 is a diagram representing motion compensation vector calculations for retrieval of data from full frame buffer memory for luminance of Fig. 8 or 4:2:2 chrominance data of Fig. 9.

Fig. 12A is a diagram, similar to Fig. 12, representing motion compensation vector calculations for retrieval of data from full frame buffer memory 4:2:0 chrominance data of Fig. 10.

Fig. 13 is a list of equations used for DRAM address calculations by the motion compensation section of Fig. 7A.

Fig. 14 is a state diagram used by a portion of the logic of the motion compensation section of Fig. 7A.

Fig. 15 is a block diagram of the prediction filter portion of the motion compensation section of Fig. 7A.

Fig. 15A is a block diagram of the reconstruction portion of the motion compensation section of Fig. 7A.

Fig. 16 is an example of an address structure for addressing a byte or relative base address of data in the memory illustrated in Figs. 8-12A.

#### **Detailed Description of the Preferred Embodiments:**

One embodiment of the present invention is for use in high definition television (HDTV). Fig. 1 diagrammatically represents an HDTV receiving and audio and video presentation system 30, which includes a signal processor and controller unit 31 having a program signal input 32 in the form of an antenna, a cable, CD ROM or other medium through which an MPEG-2 digital input signal is received, a control input from a control input device 33 through which a user makes program and presentation format selections, which may include interactive communications, a video output which connects to a video display or video presentation subsystem 34, and an audio output which connects to an audio amplifier and speaker system or audio presentation subsystem 35. The unit processor 31 includes a central processing unit or host CPU 36 which is programmed to process user commands from the control input device 33 and to operate a control system display 37, which displays information, menu selections and other information to the user and which may or may not also function as an input device. The unit processor 31 also includes an Application Specific Integrated Circuit or ASIC 40, which, when provided with configuration and selection information by the host CPU 36, decodes the raw digital bitstream signal from signal input 32 for output to the video and audio presentation devices 34 and 35. The unit processor 31 further includes a local system clock 41, which connects preferably

to the ASIC 40, and a buffer memory 42. The buffer memory 42 is in-line, sequential memory, such as dynamic random access memory or DRAM (such as a synchronized DRAM or SDRAM), and preferably includes a contiguous variable length buffer 44 for use by the ASIC 40 for audio signal processing.

Fig. 2 diagrammatically illustrates the configuration of the ASIC 40. The ASIC 40 is a single integrated circuit chip that is logically divided into a number of components or functions. The ASIC 40 includes a memory control and a data bus or gbus, represented by memory controller and bus component 50 in the diagram, which has at least one received data input connection and a plurality of two-way data flow connections. One representation of two-way connections internal to the ASIC 40 is a connection to static random access memory or SRAM 51 of the ASIC 40, which is diagrammatically represented for convenience as block 51 of the ASIC 40, but is actually distributed among the various decoders and other circuits of the ASIC 40. Another of the two-way connections is to a host interface unit 52 which connects externally with the host CPU 36, and another is to the DRAM module 42 which is external to the ASIC 40.

The ASIC 40 also includes a demultiplexer or DMUX 53 which has an input connected to the signal input 32 of the unit processor 31 and an output connected to the received data input of the bus of component 50. The DMUX 53 has a text output connected to a teletex processor 54, that is also provided on the ASIC 40 for processing collateral information such as closed caption script, interactive menus and other such data. The unit processor 40 further includes an audio decoder 55, a video decoder 56 and a local subpicture generating unit 57. The audio decoder 55 has an input side connected to the one of the two-way data connections of the bus of component 50 and an output connected externally of the unit processor 35 to audio presentation subsystem 35.

The video decoder 56 receives video program data via another of the two-way data connections of the bus of component 50, decodes by decompressing, transforming and otherwise processing the received video data, and sends the decoded and partially processed video picture data back through bus of component 50 to a video buffer 48 in the DRAM 42. This processing preferably includes the storing of decoded *I* picture frame data, the application of motion compensation calculations to extrapolate information from a buffered *I* picture to construct and store *P*-picture frame data, and the application of motion compensation calculations for the construction and storage of *B*-picture fields from buffered *I* and/or *P* frames and received *B*-picture data. The video decoder 56 also processes the motion compensated pictures stored in DRAM 42 for post-filtering and other post processing and for output by the video presentation system 34.

The subpicture generating unit 57 generates local picture information that includes control menus, display bar-graphs and other indicia used in control interaction with the user. A blender 58 is provided which combines the local video from the subpicture unit 57 with teletex information from the teletex processor 54, and with post-processed video program being output from the video decoder 56. The output of the blender 58 is connected externally of the unit processor 31 to the video presentation subsystem 34.

The ASIC 40 is provided with a control bus 60 to which a control port of each of the components 50-57 of the ASIC is connected. The ASIC 40 is also provided with a Reduced Instruction Set Controller or RISC 61, which serves as the local CPU of the ASIC 40. The RISC 61 controls the functions of the components 50-57 of the ASIC 40 through control data ports connected to the control bus 60. The RISC 61 has a clock input that connects externally of the ASIC 40 to the local system clock 41, and has another input connected to phase locked loop circuitry or PLLs 62 within the ASIC 40 used to time internal clock signals.

An example of a video signal that the system 30 must decode is a two dimensional picture made up of blocks of 64 pels in an 8 x 8 square array as is illustrated diagrammatically in Fig. 3. At the transmitter, the data representing these pels is coded as a block by application of, for example, a Discrete Cosine Transform (DCT), which is decoded as a block at the receiver of the system 30. In Fig. 3, the luminance and chrominance are represented for a fully interleaved full frame picture, with each pel represented by a luminance value  $Y$ , designated  $Y_{0,0} \dots Y_{7,7}$  for the various rows and columns of the block, each of 8 bits, or 1 byte. Where the coding is of the CCIR-601 4:2:0 chrominance format, as illustrated in Fig. 3, 2:1 horizontal and vertical subsampling has produced a signal having one chrominance data pair for each 2x2 group of 4 luminance values  $Y$ , for a total of 16 chrominance data pairs per block. The CCIR-601 4:2:0 chrominance format is the format expected to be found in broadcast HDTV. For this format, each chrominance data pair is represented by a chrominance pair value  $C$ , designated  $C_{0,0} \dots C_{6,6}$ . Each of the chrominance data pairs includes an 8-bit blue chrominance value  $u$  and an 8-bit red chrominance value  $v$ . As a result, each 2x2 group of four pels is represented by 32 bits of luminance data and 16 bits of chrominance data, or 64 bytes of luminance data and 32 bytes of chrominance data per 8x8 pel block.

Where the coding is of the CCIR-601 4:2:2 chrominance format, as illustrated in Fig. 3A, 2:1 horizontal and 1:1 vertical subsampling produces a signal having one chrominance data pair for each 2x1 group of 2 luminance values  $Y$ , for a total of 32 chrominance data pairs per block. Each chrominance data pair is represented by a chrominance pair value  $C$ , designated  $C_{0,0} \dots C_{7,6}$ . Each of the chrominance data pairs includes an 8-bit blue chrominance value  $u$  and an 8-bit red chrominance value  $v$ . As a result, each 2x2 group of four pels is represented by 32 bits of luminance data and 32 bits of chrominance data, or 64 bytes of luminance data and 64 bytes of chrominance data per 8x8 pel block. Some digital video discs (DVDs) and professional applications are expected to employ this format.

Where the picture is intended for interleaved display, that is by the alternate display of separate top (even line) and bottom (odd line) fields, to be interleaved upon display as a full frame picture, the even or top field includes the even rows of luminance values  $Y_0, Y_2, Y_4$  and  $Y_6$  and the even rows of chrominance values  $C_0$  and  $C_4$ , while the odd or bottom field includes the odd rows of luminance values  $Y_1, Y_3, Y_5$  and  $Y_7$  and the even rows of chrominance values  $C_2$  and  $C_6$ , as illustrated in Fig. 4, with each chrominance pair corresponding to a 2x2 group of four luminance values of the same field. With CCIR-601 4:2:2 chrominance format, only 2:1 horizontal subsampling is used, so that the data also contains the odd rows of chrominance data  $C_1, C_3, C_5$  and  $C_7$  (Fig. 5), yielding a signal having one chrominance data pair for each pair of two luminance values  $Y$  of the same line, for a total of 32 chrominance data pairs per block. Each chrominance data pair is accordingly represented by a chrominance pair value  $C$ , designated  $C_{0,0} \dots C_{7,6}$ . As a result, each 2x2 group of four pels is represented by 32 bits of luminance data and 32 bits of chrominance data. The even or top field includes the even rows of luminance values  $Y_0, Y_2, Y_4$  and  $Y_6$  and the even rows of chrominance values  $C_0, C_2, C_4$  and  $C_6$ , while the odd or bottom field includes the odd rows of luminance values  $Y_1, Y_3, Y_5$  and  $Y_7$  and the even rows of chrominance values  $C_1, C_3, C_5$  and  $C_7$ , as illustrated in Fig. 5, with each chrominance pair corresponding to two luminance values of a line of the corresponding field.

In Figs. 3, 3A, 4 and 5, the grid boxes group the luminance values to which each of the chrominance values in the boxes correspond.

The buffer memory map according to the preferred embodiment of the present invention is described in the context of an HDTV program for producing video frames, for example as represented by the picture in Fig. 6. The memory map is illustrated for a picture made up of an array of pels 720 wide and 480 high, grouped as a 90x60 array of 8x8 blocks of pels, or 45x30 16x16-pel macroblocks. However, the pictures may alternatively be in the form of, for example, an array of pels 1280 wide and 1024 high or some other such format. A 160x128 pel array picture would, for example, be made up of a 160x128 array of 8x8 blocks of pels, or 80x64 16x16 pel macroblocks of four 8x8 pel blocks each.

The 720x480 pel picture represented in Fig. 6 includes an array of 30 rows, each of 45 macroblocks  $S$ , which are labeled  $S_{0,0}$  through  $S_{0,44}$  for the topmost row through  $S_{29,0}$  through  $S_{29,44}$  for the bottommost row. Each macroblock  $S$  is formed of four 8x8-pel blocks. The blocks within the four macroblocks of a 2x2 square of macroblocks are further labeled  $A-D$ ,  $E-H$ ,  $I-L$  and  $M-P$  according to a repeating pattern, for each 2x2 macroblock square, to facilitate the description of the mapping of the order of the blocks in the DRAM 42. This labeling order is used below in the description of the memory map that embodies principles of the present invention.

The information flow and memory configuration that employs the memory map according to the preferred embodiment of the present invention is illustrated in the simplified diagram of Fig. 7. In Fig. 7, the raw signal of the incoming bitstream at input 32 is demultiplexed in the DMUX 53 and sent through the data bus of component 50 to the video decoder 56. The decoder 56 processes the signal in a decompression and transformation section 71, which extracts information from the various headers for interpretation by the RISC 61. The interpretation may involve storage of information to SRAM 51 on the ASIC 40 or to a portion 72 of the DRAM 42 reserved for the storage of headers, miscellaneous parameters and data. For example, information in the Video Sequence Header may contain picture size and aspect ratio data that might be used to by the ASIC to reassign memory segment labels to the video buffer 48 of DRAM 42 in accordance with the scheme of Fig. 6 and to keep track of the size of the buffer memory 48 needed for data buffering. This Video Sequence Header, for example, tells the RISC 61 whether a CCIR-601 formatted program is being received, and what that format is.

As groups-of-pictures (GOPs) are received, presentation delay requirements and other data relevant to the entire GOP are extracted from the GOP header and interpreted by the RISC 61, which responds appropriately. As each picture of the GOP is received, a picture header is extracted and interpreted by the RISC 61 to determine the picture type (e.g.,  $I$ ,  $P$  or  $B$ ), the display order of the picture relative to other pictures or its relative place in the stream, and relevant other information concerning the picture. In response to information in these headers, the RISC 61 controls the video decoder 56 and particularly the decompression and transformation section 71, as well as the memory controller of component 50 to store the picture data from the incoming pictures of the GOP which is carried out in accordance with principles of the invention.

In particular, the section 71, under control of the RISC 61, tends to the decoding and buffering, in a reference picture buffer 73 of the video buffer 48, of full frame reference pictures, either in the form of an  $I$ -picture, which is stored without reference to data from previously received pictures, or of a  $P$ -picture, which is stored after copying data from a previously received  $I$ - or  $P$ -picture. Such a reference picture is stored as a fully interleaved, that is full-frame, picture containing the alternating even and odd scan lines of the frame in the top



to bottom order in which they will appear in the displayed picture. In the preferred embodiment, two portions of the buffer 73 are provided, portions 73A and 73B, one for storing the previous full frame picture in the form of an *I* or a *P*-frame, and the other for storing the next full frame picture in the form of a *P* frame. Such *P*-frames, may be written to this memory by copying, with or without motion compensation, portions of the previous picture from the other portion 73A, 73B of the buffer 73. When a further *P*-frame is received, it will over write one of the frames in the memory portions 73A or 73B, typically the earlier of the two in presentation order, to become the next, "next" frame, while the frame in the other portion 73A or 73B of the buffer becomes the next "previous" frame. When a further *I*-frame is received, it will over write one of the frames in the memory portions 73A or 73B, typically the frame in the portion 73A. Each of the buffer memory portions 73A and 73B are further divided into halves 73L and 73C, each to store the respective luminance data and chrominance data for the picture being stored.

The decoding and storage of the picture data involves the processing of Slices of consecutive variable length strings of macroblocks (MBs) and storing them in specific memory locations assigned to the specific MBs in accordance with a memory map of described in more detail below. A Slice Header, is read and interpreted, which contains information of the vertical position of the Slice within the picture, its quantization scale and other information that may vary from Slice to Slice. MB headers for each macroblock are similarly read, which contain information of the horizontal position of the first MB of a Slice, relative to the left edge of the picture, and a relative address increment for subsequently received MBs of the same Slice. This operation places each macroblock of the picture and the data of each of its four 8-pel by 8-pel blocks in memory positions corresponding to their positions in the picture which are diagrammatically illustrated, for example, in Fig. 3A.

The DRAM 42, as illustrated diagrammatically in Figs. 8 and 11, is preferably 16 Mbits (2 MBytes) in size for systems for reproducing 4:2:0 chrominance format pictures. For 4:2:2 chrominance pictures, 32 Mbits is preferred. The DRAM 42 is preferably configured in 16 bits in width, illustrated as logical rows 74-0 through 74-15, in 32-bit (4-byte) wide columns 81, and in sequential pages 75, preferably as large as practical. The DRAM 42 is preferably addressable 64 bits at a time, via a data bus that is 64 bits wide, more particularly as described in commonly assigned and copending U.S. patent applications serial nos. 08/846,590 and 08/846,941, filed April 30, 1997, invented by Edward J. Paluch, hereby expressly incorporated by reference herein. Currently, practical sized pages 75 of currently commercially available DRAM are  $2^{12}$  (or 4096) bits in size, or 512 bytes. The bytes 76 of the reference picture buffer 73 each hold either one 8 bit luminance value or one 8-bit of chrominance value of a two chrominance value pair. The blocks of the reference *I* and *P* pictures are each mapped into the buffer portions 73A, 73B in the sequence represented by the alphabetical order corresponding to the blocks illustrated in Fig. 6, resulting in a DRAM map for the reference picture buffers 73A and 73B that is illustrated in Fig. 11.

The mapping of the screen data to the buffers 73A, 73B involves locating the datum for the upper left pel of the screen at a base for the respective buffer 73A or 73B on the first row 74-1 of the memory 42, followed by the data for the remaining seven pels of the block, which is block *A* of macroblock  $S_{0,0}$  of the picture. Each of the remaining seven lines of block *A*, which are 8 bytes of data each, from top to bottom of the block, are stored in consecutive address locations of the memory, respectively in memory rows 74-2 through 74-8. Next are stored the eight lines of the next block below the first block, which is block *C*. This fills the 16-row width

of the memory 42. For the full frame I or P pictures in buffer portions 73A and 73B of buffer half 73L, the storage of luminance data of the picture will include the data of two vertically adjacent blocks, for example blocks A and C, of the same macroblock  $S$ , which will occupy memory occupying address locations 0h to 7Fh. Next, the two vertically adjacent blocks, for example blocks B and D, that are immediately to the right of the  
 5 and horizontally adjacent to the previously stored blocks A and C are stored across the width of the buffer beginning at an address 80h. This pattern is followed with each horizontally adjacent pair of two vertically adjacent blocks being stored at increments of 80h addresses until the right edge of the screen is reached. As a result, data for the pixels across each scan line of the uppermost row of macroblocks  $S_{0,0}$  through  $S_{0,44}$  of the picture are adjacent in the same horizontal line 74 of the buffer memory, as illustrated in Fig. 11.

10 Then, the next row of macroblocks, macroblocks  $S_{1,0}$  through  $S_{1,44}$ , are stored. Because each macroblock occupies 256 bytes of data, with 512 byte size pages 75, the luminance data for two macroblocks is stored in each page 75 of the memory 42.

The RISC 61 controls the base addresses of each of the portions of memory 73A and 73B and the halves thereof 73L and 73C. In Fig. 8, the storage locations 0h and 80h are relative to base addresses for the  
 15 luminance and chrominance data buffers for the frame. These base addresses can be changed by the RISC 61 based on the storage requirements needed for the various pictures or picture formats. With the CCIR-601 4:2:2 chrominance format, the sizes of the luminance data buffers 73L and the chrominance data buffers are the same (see Figs. 8 and 9), since there are half as many chrominance value pairs  $C$  as there are luminance values in 4:2:2 format (see Fig. 3A), but two one-byte luminance values for each two byte chrominance data pair. The per  
 20 luminance data  $Y$  are stored in each of the portions 73L of the buffer 73 with the blue and red chrominance values  $u$  and  $v$  of each chrominance pair  $C$  stored in a chrominance buffer 73C.

When a P-picture or a B-picture is received, construction of a full frame or field involves the extrapolation or interpretation of the variable data of the P- and B-pictures and the calculation of Motion  
 25 Compensation vectors that specify the relocation of macroblocks from one or both reference frames stored in the reference picture buffer 73. In the case of P-pictures, the full frame P-picture is constructed and stored in one of the portions 73A, 73B of the buffer 73 from data copied in part from the previous I or P-picture in the other portion 73B, 73A of the buffer 73. The motion compensation processing is handled by a motion compensation portion 77 of the hardware of the video decoder 56.

In the case of B-pictures, the one field of the picture is reconstructed at a time and stored in one of two  
 30 overlapping halves of a B-picture buffer 78. The two halves of buffer 78, halves 78A and 78B, respectively store the alternate top and bottom fields of the next picture to be presented to the output device. The buffer 78 is configured as a circular buffer with the two halves 78A, 78B thereof overlapping so that the size of the buffer 78 is optimally approximately 0.53 as large as the full-frame portions 73A, 73B of the buffer 78. The 0.5 allows  
 35 for the storage of one full field, which is half of a full frame, where the 0.03 allows for the storage of the data for one row of macroblocks of one field of the picture. When either of the buffer halves 78A or 78B contains a complete reconstructed field, its size will be 0.50 of a frame, and the buffer 78A or 78B containing the field will store the field data until the presentation time stamp (PTS) for that field equals the time in the system time clock, indicating that the field is ready for display. When this occurs, the field data will be read from the buffer 78A or 78B, by scanning horizontally from top to bottom, with the scanned data being processed by post

filtering section 79 of the video decoder and sent to the video presentation device 32 for display. The preferred operation of the post-filtering section 79 is described in U.S. patent application serial no. 08/\_\_\_\_,\_\_\_\_ filed November \_\_, 1997, by the inventors hereof and entitled Letterbox Filter Apparatus and Method, hereby expressly incorporated herein by reference.

5           When the presentation of one reconstructed B-picture field from buffer 78 is begun, the decoding and storing of the next field is simultaneously triggered. As the first horizontal row of macroblocks from the stored field is being post-filtered and output, the decoding of the first horizontal row of macroblocks from the other field is reconstructed and stored in the remaining 0.03 frame size portion of the memory 78. When the first row of macroblocks has been output and the output of the second row of the first stored field is begun, the second  
10          row of macroblocks from the other field is stored to the memory 78, overwriting the first row of macroblocks of the first field. The buffer 78 may be made larger, for example to correspond in size to about 0.56 or 0.59 of the size of the full frame buffers 73A and 73B, to respectively hold two or three additional rows of macroblocks rather than one, to allow greater flexibility in this reading and writing of the data of consecutive alternating fields.

15           The bytes 76 of the reconstructed B-pictures that are stored in the buffer 78 also hold one 8-bit luminance or chrominance value. The reconstructed blocks of the B-pictures are mapped into the buffer 78A or 78B in the sequence represented by the alphabetical order corresponding to the blocks illustrated in Fig. 6 that results in a DRAM map for the reference picture buffer 78 that is illustrated in Fig. 8A. The order of storage of the screen data in the buffers 78A, 78B also involves beginning at the upper left corner of the screen and  
20          storing each odd or even line of a block, which is 8 bytes of data, from top to bottom of the block, followed by the lines of the next block below the first block until the width of the buffer, 16 rows, is reached. In Fig. 8A, only the even field in buffer 78A is illustrated. For the odd field in buffer 78B, the leftmost subscripts will be 1, 3, 5 and 7, indicating the odd rows of the blocks. Because only one field of data is being stored in the buffer 78A or 78B, only half as many rows are stored, so that the odd or even rows of four blocks can be stored across  
25          the 16-row width of the buffer memory 42. For the field sized B-pictures in buffer 78, the luminance data for four vertically adjacent blocks, for example blocks A, C, I and K of two vertically adjacent macroblocks S, will be stored across the 16-bit width of the memory. The four vertically adjacent blocks, for example blocks B, D, J and L, immediately to the right of and horizontally adjacent to the previously stored blocks are stored across the width of the buffer at an address 80h greater than the storage location of the first four blocks. The overall  
30          layout of the memory for buffer 78L luminance and for buffer 78C in the case of 4:2:2 formatted chrominance is illustrated in Fig. 11A. As a result, data for the pixels across a scan line of the picture are adjacent in the same horizontal line 74 of the buffer memory for B-pictures as well as for the I and P-pictures.

          The RISC 61 controls the base addresses of each of the portions of memory 78A and 78B and halves thereof 78L and 78C, which contain the respective luminance and chrominance data for reconstructed B-  
35          pictures. In Figs. 8A and 9A, the storage locations 0h and 80h are relative to the base addresses. These base addresses can be changed by the RISC 61 based on the storage requirements needed for the various pictures or picture formats, and in fact rotate through the buffer 78 as one field overwrites the other. With the CCIR-601 4:2:2 chrominance format, the sizes of the luminance data buffer 78L and the chrominance data buffer 78C are the same, since there are two one-byte luminance values for each two byte chrominance data pair. The pel

luminance data  $Y$  are stored in each of the portions 78L of the buffer 78 with the blue and red chrominance values  $u$  and  $v$  of each chrominance pair  $C$  (Fig. 5) stored in a chrominance buffer 78C.

With 4:2:0 chrominance format, the chrominance data for a frame or field is one half of that for chrominance data for 4:2:2 chrominance format, and one half that of the luminance data for the frame or field. In this case, which is the expected case for broadcast HDTV, the chrominance frame data will be stored as illustrated in Fig. 10. This is the same mapping that is used for one field of luminance, which accordingly indicates that the chrominance buffer portions 73C of frame buffers 73A and 73B are each 0.5 the size of frame buffers 73A and 73B. The overall memory map for the chrominance buffer portions 73C are the same as for one field's worth (left portion) of the  $B$ -frame buffer illustrated in Fig. 11A. The  $B$ -picture chrominance data of 4:2:0 chrominance format illustrated in Fig. 10A, with eight vertically adjacent blocks or four vertically adjacent macroblocks extending across the 16-row width of memory 42. The overall arrangement of the memory for 4:2:0 chrominance data in the  $B$ -field buffers 78A and 78B is illustrated in Fig. 11B.

An advantage of the memory map of Figs. 8-8A, 9-9A, 10-10A and 11-11B is the decreased likelihood that any given attempt to retrieve or store a macroblock of the reference picture from or to the reference picture buffer 73 will require the crossings of a page boundary. Page boundary crossings impose a substantial time penalty on the processing of the data, so any elimination of page boundary crossings in the retrieval of data is highly desirable. The alphabetically designated storage order of the blocks provides a statistical reduction in the likelihood that a request to retrieve macroblock data will cross a page boundary in the memory. In addition, the storage arrangement provides further reduction in the page crossing likelihood where the block-to-page ratio were to increase through the use of a more efficient data format or as memory of increased page size becomes available. Further, the arrangement of data so that the same lines of adjacent blocks are adjacent on the same line of memory increases the efficiency of data storage and retrieval. Also, provision of chrominance data of the same pixel in close proximity in memory, so that a single memory read will retrieve all relevant chrominance data for the pixels, increases the efficiency of memory storage and retrieval as well as the efficiency of the use of the processing capacity of the ASIC 40. Scan line output from both the frame buffers 73A and 73B and the field buffers 78A and 78B can be made by merely incrementing the address of the data by 80h increments in the memory controller of component 50.

The motion compensation hardware 77 constructs a video frame or field from received  $B$ -picture data and information stored in the  $I$ - and  $P$ -picture buffer 73. The reconstructed video frame or field is stored in a video output buffer 78 in DRAM 42 for post-filtering by post-filter hardware 79 and buffered output to video presentation system or display 34. The storage of the reconstructed picture for display at two interleaved fields is provided, with the video output buffer 78 being divided into the two halves, buffer 78T, which stores the top or even field data of a frame, and buffer 78B, which stores the bottom or odd field data of a frame.

Video output and post-filtering can take place from either  $B$ -field buffers 78A, 78B or from the  $I$  or  $P$ -frame buffers 73A, 73B. The output and post-filtering from  $I$  or  $P$ -frame buffers takes place one field at a time, with odd or even rows being read from the buffer 73, depending on whether bottom or top field is being filtered and displayed. With the video output either buffer 73 or 78, mapped as illustrated and described above, the output of the data across lines of display follows rows 74 of the memory, which can be made optimum memory reading and addressing and with a minimum of page boundary crossings. The optimal reading occurs in 64 bit



memory read cycles in which data is read from the memory that is illustrated in Figs. 8, 8A, 9, 9A, 10 and 10A, in rectangular array. For example, corresponding lines of data of blocks *A*, *B*, *E* and *F* of horizontally adjacent macroblocks lie in the same line of the picture, with the data of lines of blocks *I*, *J*, *M*, and *N*, or of blocks *C*, *D*, *G* and *H*, or of blocks *K*, *L*, *O* and *P*, respectively, lying on the same lines of the picture. Accordingly, the storage and retrieval of reconstructed data to the buffers 73 and 78 are made with reduced page boundary crossings, thereby improving the overall processing efficiency of the receiver. In addition, one of the fields can be output from one of the buffer halves 78T or 78B while a field is being constructed in the other buffer half.

The manner in which the memory map, according to the principles of the invention, provides an economy of page crossings, or rather provides a low ratio of pages per data retrieval or storage requirement from and to the DRAM 42 is illustrated in Fig. 12 for reads and writes from and to a full frame *I* or *P*-frame luminance buffer 73L (or 4:2:2 chrominance buffer 73C), and in Fig. 12A for reads and writes from full frame chrominance buffer 73C (for 4:2:0 chrominance. This is particularly beneficial with reads from the reference picture buffers 73 because the motion compensation vectors can specify any pel or half pel location as that of the 16x16 pel data to be copied from a reference picture. (For *B*-pictures in field buffers 78, blocks are only written and then only to the original macroblock base addresses.

As explained in detail in the Paluch applications incorporated by reference in the discussion above, the preferred DRAM 42 is arranged in 16 logical rows 74 and in 32-bit wide columns 81, so that the intersection of one row and one column includes four 8-bit bytes of information, such as four luminance values (for four horizontally adjacent pels) or four chrominance bytes (or two 2-color chrominance value pairs). For fully interleaved frames of video data, a rectangle of eight rows by four 32-bit columns in the full frame picture buffers 73 define either the luminance or the chrominance (in 4:2:2 format) for an 8x8 pel block of a picture or two vertically adjacent 8x8 pel blocks of such data for the even or odd field in field buffers 78. Accordingly, for fully interleaved frames of video data, a rectangle of sixteen rows by eight 32-bit columns in the full frame picture buffers 73 such data for a 16x16 pel macroblock *S* of a picture, or two vertically adjacent 16x16 pel macroblocks for the even or odd field. Further, as explained in detail in the Paluch applications, the memory controller of component 50 can address logical rectangles of data in contiguous (frame mode) or alternate (field mode) rows or adjacent columns, up to the 16-row width of the memory of up to a predetermined number of columns.

The memory mapping described above facilitates the efficient and rapid storage and retrieval of data in the DRAM 42 in connection with two primary functions of the video decoder 56. These functions include the temporal filtering function of motion compensation, in which reference macroblocks of *I*- or *P*-picture data are read from one or both of the full frame buffers 73A, 73B and reconstructed extrapolated or interpolated macroblocks are written either to *P*-picture frames in buffer 73 or *B*-picture fields in buffer 78 for output after or between the presentations of the reference frames. Because motion compensation calculations employ half-pel interpolation in the reconstruction process, such data transfers involve the retrieval and storage of 17x17 pel blocks of data based at any pel location on a picture. With 32-bit wide data reads and writes as described above, this calls for the retrieval and storage of 17x20 pel blocks of data. The present invention facilitates the storage and retrieval of such data blocks without an excess of page crossings, that is, without the need to address a large number of pages to perform such read and write operations.

These facilitated functions of the video decoder 56 also include the spacial post-filtering function and the outputting of the fields of data from the frame or field buffers 73 or 78 for transmission to the display 34. This post-filtering and outputting involves the reading of one or more adjacent or alternate scan lines of pixel data across the entire width of the screen, for example, across 720 pixels. Typically, only one or two lines of data are so read. With the present invention, the data are also read from the scan lines with a relatively small number of page crossings. The present invention optimizes the reading and writing of data for both the motion compensation functions of the video decoder 56 and for the post-filtering and output functions of the decoder 56.

Figs. 12 and 12A are illustrations of the memory maps according to one embodiment of the invention showing the layout of the pages 75 of the memory 42 in an array that corresponds to that of a picture. In Fig. 12, the pages 75 are arranged according to the two dimensional layout of the luminance data for frame pictures stored in the buffers 73L, while in Fig. 12A, the pages 75 are arranged according to the two dimensional layout of the 4:2:0 chrominance data for frame pictures stored in the memory buffers 73C. In Figs. 12 and 12A, the individual 8-bit bytes 76 or pel data values are not illustrated, but the 32-bit (4-byte) columns 81 are differentiated by the light vertical lines 82, with the rows 74 being differentiated by the light horizontal lines 83. The macroblocks  $S$  of 16x16 pels are differentiated by the medium weight vertical lines 84 and in Fig. 12A also by medium weight horizontal lines 85. Adjacent memory pages 75 are separated by the heavy vertical lines 86. The memory controller of component 50 is capable of addressing rectangular arrays of adjacent columns 81 and adjacent or alternate rows 74, which arrays can cross page boundaries 86 between horizontally adjacent memory pages, although separate memory address operations are required to read or write each horizontally adjacent page that is included in such a rectangle of addressed data. Double horizontal lines 87 are used in the drawing to differentiate vertically adjacent macroblocks  $S$  that lie on different non-adjacent pages 75 of the buffer memory 42. The memory controller of component 50 cannot address a rectangle of data that crosses such a boundary 87, since the pages are not physically adjacent in the memory 42, but only the data on these pages happens to represent pixels from macroblocks that appear adjacent on the display.

An example of how motion compensation is carried out with the memory map in Fig. 12 where the luminance or 4:2:2 chrominance data of a full frame macroblock  $S_p$  of a  $P$ -picture, say in buffer 73B, is to be rewritten with a 16x16 pel square 90 (superimposed on Fig. 12 for illustration) of data from a reference  $I$ - or  $P$ -picture that is displaced, as specified by a motion compensation vector 88, at some relative vertical and horizontal distance in buffer 73A. The motion vector 88 may specify, for example, that the 16x16 pel square 90 that is to be copied is located, for example,  $25\frac{1}{2}$  pels below and 23 pels to the right of the macroblock to be reconstructed. In general, retrieval of the specified 16x16 pel square 90 from DRAM 42 may occur as a 16x16 pel square or as rectangles of 17x16 pels, 16x20 pels or 17x20 pels. Retrieval of only the 16x16 pel square 90 is possible where the vector 88 calls for vertical displacement of an even number of pels and horizontal displacement of a number of pels that is divisible by four. Vertical displacement of a half pel is met with retrieval of a 17 row rectangle where horizontal displacement of a number of pels that is not divisible by four or that includes a half pel is met with retrieval of a 20 byte (5 column) wide rectangle.

In the case of the square 90, which is displaced  $25\frac{1}{2} \times 23$  pels, retrieval of the data to be copied from the reference picture in buffer 73A occurs with the memory controller of component 50 addressing a 17x20 pel

rectangle 92 of data originating at base address 91 of the buffer 73A in the DRAM 42. This retrieval uses two requests for rectangular arrays of data, since the horizontal boundary 87 divides the logical rectangle to be retrieved into two physically non-contiguous rectangles of data, one on pages 75<sub>69</sub> and 75<sub>70</sub> and one on page 75<sub>47</sub>. These two requests are managed by the motion compensation logic 77 of the video controller 56. The retrieval operations for rectangle 92 from buffer 73A involve retrieval from three memory pages, pages 75<sub>47</sub>, 75<sub>69</sub> and 75<sub>70</sub> of the buffer 73A, while the write operation of macroblock  $S_p$  to buffer 73B involves a writing to only one page, page 75<sub>24</sub> of the buffer 73B. After half pel interpolation is performed by the video decoder 56 on the retrieved data, the 16x16 pel macroblock of data  $S_p$  is extrapolated from the copied data and the reconstructed macroblock is written to buffer 73B at base address 94.

With the retrieval of a rectangle of luminance data for construction of a  $B$ -field in buffer 78, where a macroblock  $S_B$ , for example having a base location 99 in buffer 73B, is to be reconstructed, the data to be copied from reference pictures may be specified by two motion vectors, one motion vector such as the vector 88 of Fig. 12 indicating the use of data from 16x16 pel square 90 of data from a reference  $I$ - or  $P$ -picture in buffer 73A, and in addition by a similar vector, for example vector 93 indicating the use of data from another 16x16 pel square 98 of data from a reference  $P$ -picture in buffer 73B (also superimposed on Fig. 12 for purposes of illustration) that is located, for example, 8 pels above and 7 pels to the left of the macroblock  $S_B$  to be reconstructed. The reconstructed data is produced by interpolation of the data retrieved from the two squares 90 and 98. To retrieve the data to be copied from the reference picture in buffer 73A, the memory controller of component 50 will address the 17x20 pel rectangle 92 of data originating at base address 91 of buffer 73A. To retrieve the data to be copied from the reference picture in buffer 73B, the memory controller of component 50 will address the 17x20 pel rectangle 95 of data originating at base address 96 of the buffer 73B in the DRAM 42. After half pel interpolation is performed by the video decoder 56 on the retrieved data, the 16x16 pel macroblock of data  $S_B$  is reconstructed by interpolation between the two retrieved reference rectangles 92 and 95 and written to buffer 78A at address 97 relative to the base address of the buffer 78A. Since the  $B$ -picture to be reconstructed is a field picture, only alternate odd or even lines of the rectangles 92 and 95 are retrieved. The memory controller of component 50 is able to address rectangles of data from either alternate odd or even lines only. In the writing process for a macroblock of field data, only eight corresponding odd or even lines are written to buffer 78. The retrieval operation of rectangles 92 and 95 from buffers 73A and 73B each involve retrieval from three memory pages, while the write operation of macroblock  $S_B$  to buffer 78A involves a writing to only one page. With pages of the size and configuration provided by the preferred memory of the present invention, no macroblock of data in either the reference frame buffer 73 or in the field buffer 78 extends onto more than one page.

The page crossing reduction is realized to an even greater degree with the retrieval of the corresponding 4:2:0 chrominance data. As Fig. 12A demonstrates, a rectangle of 4:2:0 chrominance data has one half the vertical height, or occupies one half the number of rows, as does the luminance data. Thus, the need for the motion compensation section 77 to divide the retrieval request to two instructions to the memory controller of component 50 in order to avoid the crossing of a horizontal boundary 87 between vertically adjacent macroblocks is even less likely to occur.

Further memory addressing and page crossing advantages are realized in connection with the post-filtering and output functions, where full picture width rows of data for a single or for two adjacent or alternate scan lines are retrieved from buffers 73 or 78 for post-filtering and display. The post-filtering operation includes the interpolation of data to change the aspect ratio of a picture when the received images have a different aspect ratio than the display, and to convert the chrominance format into three color signals per pixel for display. With the memory map of the described embodiment of the invention, all data of a given scan line of a picture are stored in a continuous sequence on the same row of the memory 42, and are accordingly addressable with a single addressing step by the memory controller of component 50, subject only to the maximum size of data request that the controller of component 50 is able to handle. This maximum size is, for example, 88 bytes 76, which is 22 columns 81. Similarly, two adjacent lines of a frame or field are stored on adjacent rows of memory, up to the 16 row width of the memory. Thus, in the frame buffers 73A, 73B the 16 rows of pel data for a horizontal row of adjacent macroblocks are stored in 16 continuous rows in memory. In the field buffers 78A, 78B, the odd or even rows of two vertically adjacent rows of macroblocks are stored in 16 continuous rows of memory.

With the invention, one instruction to the memory controller of component 50 for the output of a scan line of the screen is all that is required for the memory controller of component 50 to issue separate read addresses, each incremented by 80h. When a single scan line is read for post-filtering, a data request is made by the memory controller of component 50 for 22 adjacent 32-bit (4-byte) wide elements of data from a single row 74. The memory controller of component 50 handles this request and reads data in a stream from the same row 74 across boundaries 86 and from three or four horizontally adjacent pages of memory, so that the post-filtering section 79 can process four bytes of horizontally adjacent pixel values on a first-in first-out basis. More than on row, for example two rows, can be read in a 44 byte wide and two byte (row) high rectangle. Such multiple rows are read column by column, in zig-zag fashion, returning first 32-bits (4-bytes) from one row, then the four vertically adjacent bytes from the same column 81 in the next row of the frame or field being read, then two sets of four bytes from the same two rows of the next column to the right, across boundaries 86 and from two or three horizontally adjacent pages 75. Crossings of boundaries 87 between vertically adjacent pages would occur only where the post-filtering section 79 of the video decoder 56 were to request data from multiple rows than lie on opposite sides of such a boundary, which can be controlled or avoided by the logic of the post-filtering section 79.

The above description of Fig. 12 has, for simplicity, assumed only a single portion of the respective buffers containing either the luminance or chrominance data. The reading and writing operations and memory addressing discussed is duplicated for both portions of the buffers, that is, for each the luminance and chrominance data. The chrominance macroblocks are the same size as the luminance macroblocks for 4:2:2 chrominance format but are 1/2 and twice the size of the luminance macroblocks for 4:2:0 and 4:4:4 formats, respectively, which affects the number of page crossings when the chrominance data is addressed. Fig. 12A is a representation of 4:2:0 chrominance motion compensation memory addressing for frame picture storage in buffers 73.

The generation of addresses for retrieving macroblocks *S* of data from the *I* and *P* frame data in the buffers 73A and 73B is carried out in the first instance by the motion compensation section 77 of the video



decoder 56, the logic blocks of which are illustrated in Fig. 7A. Referring to Figs. 7 and 7A, when a frame of data is received at 32 by the DMUX 53, the data passes through the data bus of the bus and memory controller 50 and, under control of memory controller 50a thereof. After various headers are processed, the flow of frame data ultimately involves the transmission of data for series of macroblocks, each of which contains a macroblock header and either a stream of compressed and transformed pixel data or a motion vector specifying the relative location on the screen from which data is to be copied, extrapolated or interpolated from one or two frames of data in *I* or *P* pictures stored in one or both of the buffers 73A, 73B.

When the picture is an *I* picture, the data is all pixel data, which is decompressed and transformed by the decoder portion 71 of the video decoder 56 and passed through the motion compensation section 77 through the bus 50b to one of the buffers 73A or 73B where the macroblocks thereof are stored at their appropriate locations as part of a full frame picture. When the picture is a *P* or *B* picture, the macroblock data can be either pixel data or motion compensation vector data indicating that the pixel data for the macroblock is to be copied from a macroblock sized square of a reference picture stored in one or both of the buffers 73A, 73B. If a macroblock of data received for the *P* or *B*-picture is new pixel data, the new data is stored. For *P* pictures, the data is stored in the appropriate macroblock location in a frame 73A or 73B of the reference buffer 73. For *B*-pictures, the data is stored in the appropriate field portion 78A or 78B of the output picture buffer 78.

If, for either a *P*-picture or a *B* picture, a motion vector appears in the data, the decoder logic 71 removes the motion vector data from the stream and passes it to an address generator and control block 111 of the motion compensation logic 77. If the motion vector is, for example, the motion vector 88 of Fig. 12, the block to be copied will be the block 90, requiring the reading from a frame buffer of buffer 73 the rectangle of data 92, which lies on three separate pages 75 of memory, pages 75<sub>47</sub>, 75<sub>69</sub> and 75<sub>70</sub>. Because this rectangle 92 is divided by a horizontal page boundary 87, the address generator 111 of the motion compensation logic 77 generates two data requests, one for the upper portion of the rectangle 92 that lies on page 75<sub>47</sub>, and one for the lower portion of the rectangle 92 that lies on pages 75<sub>69</sub> and 75<sub>70</sub>. These requests are sent one at a time to the memory controller portion of the memory controller and bus 50. The memory controller 50a will generate a single request for a rectangle of data for the upper rectangle which lies entirely on page 75<sub>47</sub>. When the upper rectangle of data has been retrieved and processed, the memory controller 50a will generate two requests for separate rectangles of data for the lower portion of the rectangle 92, one request for the rectangle that lies on page 75<sub>69</sub> and one request for the rectangle that lies on page 75<sub>70</sub>.

In response to the individual requests, macroblocks of data are returned from the buffer 73 in DRAM 42 via the bus of 50b to a half-pel filter section 112 of the motion compensation logic 77 where, along with motion vector information from the address generator 111, the half-pel interpolation, if required in either the horizontal or vertical direction, is carried out. The data from the filters 112 is then sent to a reconstruction section 113 where extrapolation or interpolation from the macroblock pixel data of one or two frames is carried out. Data of macroblocks from the input stream 32 or from DRAM 42 are respectively stored in the SRAM 51, two of the distributed portions of which are designated eMEM 51a and pMEM 51b in Fig. 7A, as explained in more detail below.

The motion compensation unit 77 applies motion compensation to macroblocks copied from the reference pictures in the reference picture buffers 73A and 73B, collects coefficients or motion vectors from the

Transform Unit 71, and generates reconstructed picture from the motion vectors and copied macroblocks according to the MPEG-2 standards. The motion compensation unit 77 maintains interfaces with the memory controller and data bus 50b so that the motion compensation unit 77 can read in the reference picture data from the DRAM 42 and store the reconstructed data into the memory 42

5 The general responsibilities of the motion compensation unit 77 include generating address and data block size requests for macroblocks of the reference picture data in buffers 73A, 73B and placing data transfer requests to the memory controller 50b. The motion compensation unit 77 also runs retrieved reference data through horizontal and vertical prediction filters. Predicted data is stored in portions of the SRAM 51 that are distributed in this unit. The unit 77 further supports half pel interpolation, that is, motion vectors having  
10 horizontal or vertical components that can take values at increments of one half pel. The unit 77 is responsible for reconstructing picture data from the half pel prediction values and the from the motion vector coefficient values supplied by the decoder 71.

The motion compensation unit 77 depends on the decoder unit 71 for the motion vectors and the address of the reference block, and for the inverse-transformed coefficient. It depends on the memory controller  
15 unit 50b for data transfers.

The address generation unit 111 accepts the motion vectors from the decoder 71 and calculates the starting address of the reference macroblock in the reference picture buffer 73. The address generation unit 11 issues OMta transfer requests to the memory controller unit 50. In response to requests, data transfer from  
20 DRAM 42 occurs in 64 bit (8 byte) chunks at addresses aligned at 8-byte boundaries. When data returns from DRAM 42, they are latched in the motion compensation unit 77. Each 8 bit element of these latched data is then run through horizontal and vertical filters, and the filtered data is stored in array in SRAM 51. Since the starting address of the reference block may not be aligned at the 8-byte aligned address, multiplexing is required at the input of the SRAM array.

For I-pictures, prediction is not required, and when an I-picture is received and passed through the  
25 motion compensation unit 77, the motion compensation hardware sits idle. For both P and B-pictures, prediction data is required for reconstruction of the picture. In cases of B-pictures, the predicted data is obtained by averaging two such predictions, one being the output of the half-pel at the time and the other being a value out of the pMEM portion of SRAM 51b. that was stored after a forward prediction. The reconstruction hardware 113 supports this averaging function of half-pel filters 112.

30 The other SRAM array 51a, designated eMEM, holds coefficient data transformed in the decoder 71. Reconstruction of the picture starts once eMEM is full. The motion compensation unit 77, at such a time, issues a data transfer request and starts the reconstruction. The reconstruction basically involves adding signed numbers from the decoder output that are stored in eMEM 51a to the half pel filter outputs stored in pMEM 51b for non-intra blocks, or blocks of B or P-pictures. For intra blocks or blocks of I-pictures, however, the addition  
35 is not required. The adder output is clipped before it is latched at the output of the reconstruction unit 113 when the hardware is used to reconstruct the picture. The calculations made to determine the addresses and the number of bytes for a data request are set forth in Fig. 13, with the interface signals for interfacing of the motion compensation section 77 with other units being as set forth in the following table, Table 1:

NAME	WIDTH	TYPE (I/O)	DESCRIPTION
gresetn	1	INPUT	GLOBAL RESET SIGNAL
ghaltn	1	INPUT	GLOBAL HALT
sclk	1	INPUT	GLOBAL CLOCK
gbus	64	OUTPUT	64 BIT GLOBAL BI-DIRECTIONAL BUS
mc_refblk_req	1	OUTPUT	REQUEST FOR REFERENCE BLOCK DATA TO THE MEMORY CONTROLLER
mc_mvreq	1	OUTPUT	REQUEST FOR MOTION VECTOR TO THE VLD
mc_mbax_req	1	OUTPUT	REQUEST FOR CURRENT MACROBLOCK ADDRESS X-DIMENTION TO THE VLD
mc_mbay_req	1	OUTPUT	REQUEST FOR CURRENT MACROBLOCK ADDRESS Y-DIMENTION TO THE VLD
vld_my_fifo_empty	1	INPUT	MOTION VECTOR FIFO IN VLD IS EMPTY - AN INACTIVE LEVEL VALIDATES THE MOTION VECTOR ON THE BUS
vld_my	1	INPUT	MOTION VECTOR DATA FROM THE VLD
vld_mbs	1	INPUT	MACROBLOCKS START SIGNAL FROM THE VLD
vld_intra	1	INPUT	VLD INDICATING THAT THE DATA IS DECODING IS FROM AN I-CODED PICTURE
vld_pic_b	1	INPUT	AN ACTIVE LEVEL ON THIS SIGNAL INDICATES A B-CODED PICTURE DATA IN THE VLD PIPELINE
vld_mv	12	INPUT	12 BIT BUS OVER WHICH MOTION VECTORS AS WELL AS THE MACROBLOCKS ADDRESS IS TRANSFERRED FROM THE VLD TO THE MC
vld_emem_full	1	OUTPUT	MOTION COMPENSATION UNIT ASSERTS THIS SIGNAL TO STALL THE COEFFICIENT DATA FLOW FROM THE IT AFTER 2/3(?) BLOCK TRANSFER
it_bs	1	INPUT	START OF A BLOCK DATA FROM THE INVERSE TRANSFORM
it_coeff_dat	9	INPUT	CO-EFFICIENT DATA FROM THE INVERSE TRANSFORM

TABLE 1

The functionality view for the transfer of reference picture data from buffers 73A, 73B of DRAM 42 is set forth in Fig. 14. In particular, as the state transitions diagram depicts, *vld-mbs* is an indication of the start of a macroblock processing. At this time, *vld-intra* is sampled to check to see if motion compensation is required for the macroblock. For coded pictures that require motion compensated reference calculation, the state machine waits until a decoder unit 71 motion vector FIFO is not empty. This control hardware of the address generation unit 111, then generates a request for a motion vector by making two consecutive requests for *X* and *Y* components of the motion vectors. Once the hardware has both components of a motion vector, the address of the reference block is calculated. The address generation unit 111 then places a request for data transfer to the memory controller unit 50.

*B* coded pictures are indicated by an active *vid\_pic\_b*. For these, another set of components for motion vectors is requested from the decoder unit 71. These are used in a similar way as the first one of vector components to generate the address for the other reference.

In response to the request for reference block data to the memory controller 50a, the memory controller 50a fetches continuous sequences of 64-bit data from 8-byte aligned DRAM addresses, and sends them to the motion compensation unit 77 over the 64-bit bus of data bus 50b, as illustrated in Fig. 15. At the input of the prediction hardware 112, these 64 bits are latched in latches 121 as eight 8-bit elements. These elements pass through both horizontal and vertical filters 122 before being stored in the pMEM or SRAM array 51b. The filters 122 pass through the input element, or an average of two adjacent elements expressed as an 8-bit quantity. The top half 123 of the logic in Fig. 15 is used for elements adjacent horizontally, while the bottom half 124 is used for elements adjacent vertically. In both cases, the least significant bit of the motion vector component determines if half-pel prediction in that particular direction is required.

Since a macroblock of data is treated eight elements at a time, the rightmost element of the previous 64-bit data is saved to supply the left adjacent element for the next set. However the leftmost column of the screen does not have left adjacent elements, so in that situation the data is passed through and ignored while storing in the SRAM 51a.

The algorithm for reconstruction performed in the reconstruction logic 113 is basically an addition in adder logic 131 of the co-efficient calculated in the Inverse Transform (IT) portion of decoder unit 71 to the predicted reference data. This result is clipped by clipper logic 132 to represent an 8-bit positive quantity. Thus, the adder adds numbers from the eMEM 51a, holding the coefficient function, and pMEM 51b, holding the predicted picture data. *I*-coded pictures do not require motion compensation, so one input to the adder is forced to zero in case of *I*-pictures. This same set of adders 131 is used to combine forward and backward prediction in the case of *B*-coded pictures. Output of the half-pel filters 112 is multiplexed at the input of the adder 131 instead of the eMEM 51a output. The result is shifted right by one bit. This gives a bidirectional prediction. The result is stored back in the RAM 51.

Further referring to Fig. 15A, the transform portion of the decoder 71 activates *it\_bs* to mark the beginning of a 64x9 block of data. After activating *it\_bs*, the decoder 71 sends streams, at a rate of one per clock cycle, of 9-bit coefficients. Eight of these are collected in latches 133 before all of them are stored in the eMEM 51a of SRAM 51. The eMEM 51a is only equipped with storage for one macroblock, for example, six



blocks. Any possibility of overflow will generate a back propagating stall to the transform portion of the decoder 71, so that they are preserved in earlier stages of the pipeline.

Applying the description of the memory map and address methodology set forth above, the addresses of each column 81 of each row 74 of the first page 75-1 of the DRAM buffer 42 is set forth in Figs. 8-12A. Each such page 75 stores 16 rows of 32 bytes each, and is thereby two macroblocks wide and one to four macroblocks high, depending on the luminance or chrominance format of the data and whether the picture is frame or field, as explained in connection with Figs. 8-11B above. The first page 75-1 is shown with a base address (upper left corner pixel) of 000000h, where the next page 75-2 is shown with a base address of 000200h. As illustrated, in the preferred embodiment, the memory addresses wrap from row 74 to row 74 after each 8 bytes (two columns) along a row. When the 16th row 74-16 is reached, the addresses jump to the first row 74-1 8 bytes (two columns) to the right. In this manner, the reading of data across the same row 74 involves the reading of 64 bits (8 bytes or two columns) followed by the incrementing of the address by 80h and the reading of 64 more bits of data.

Fig. 16 shows how a 21-bit word can address the bytes of storage of a 16-Mbit (2 MByte) DRAM 42, illustrated for the case where the base address is at the upper left corner of the picture. The two rightmost bits can specify one of the four bytes of a 32-bit column 81 of a line 74. These bits are not needed to address a 32-bit wide block of data. The third bit specifies the right or left column 81 of an 8x8 block. This bit is not needed to address a 64-bit wide block of data, which would be equivalent to one line of an 8x8 block. In the preferred embodiment of the invention, the addresses wrap after each 8 byte line segment of the picture to store 16 such lines in consecutive locations of the memory 42, or two 8x8 blocks stored consecutively line by line. This 8-byte line could be of a different size, for example, a 16 byte or pixel line, wrapping line by line, for example for a full macroblock of luminance data or two vertical macroblocks of 4:2:0 chrominance data. In this case, the 4th bit from the right would designate the left or right block of a macroblock.

The next four rightmost bits specify the one of the 16 rows 74 of the logical memory. In the preferred embodiment, these are the 4th through 7th bits. The 4th bit specifies whether the row is even or odd, and can be fixed to selectively address data from one field only when, for example, reading from an interleaved full frame buffer 73. The 8th bit specifies the left or right 8x8 pel block of a macroblock. Incrementing this bit while fixing the three bits 4-7 moves the address along the same line 74. Incrementing the 8th bit and the bits to its left effectively scans to the right across a line of the picture. The 9th bit will specify the left or right macroblock of a horizontally adjacent macroblock pair that would occupy the length of one page 512 byte page 75 of memory. Thus, the 1st through 9th rightmost bits can be used by the memory controller 50a to formulate the addressing of data stored on the same page 75 of memory, such as for reading or writing rectangular blocks of data residing on a single memory page.

The next bits, bits 10-21, can be used to specify pages of memory. The memory controller 50a can increment the right ones of these bits move across adjacent pages to scan across a horizontal picture line. The motion compensation section 77 can add to these bits to move vertically down the screen. The 9th bit would also be of interest to the motion compensation logic 77 to keep vertical alignment of the macroblocks when the number of macroblocks that extend across a picture width is odd, as in the case of the embodiment illustrated in Figs. 12 and 12A. One of the bits 10-21 can be used for bank switching. For example, if the 10th bit

- 24 -

designates one of two alternate banks of the memory 42, consecutive pages can be alternately stored in alternating banks of the memory 42.

5 The ability to address rectangular blocks of data and to break up the memory access requests between the memory controller 50a and the motion compensation logic 77, or to scan individual lines of data across a screen, with simple fixed increment addition to the address is an advantage of the memory configuration described above.

Those skilled in the art will appreciate that the applications of the present invention herein are varied, and that the invention is described in preferred embodiments. Accordingly, additions and modifications can be made without departing from the principles of the invention.

What is claimed is:

1. A buffer memory for use with a digital video decoder for decoding pictures in basic block size units of pixels, comprising:

memory means for storing at least one frame or field of picture data in a plurality of logical rows, the plurality of rows including at least as many rows as the number of pixels across one dimension of a basic block size unit; and

memory controller logic means for storing data of the picture in basic block size units of pixels mapped in an array therein with data across each scan line of the picture being stored in sequence in a single row of the memory, with data of pixels of the same basic block size unit that appear vertically adjacent in the picture being stored in a common column in adjacent rows of the memory.

2. A method of mapping a digital video data of an MPEG-2 compatible picture in a buffer memory comprising the steps of comprising:

providing a buffer memory for storing at least one frame or field of picture data in a plurality of logical rows, the plurality of rows including at least as many rows as the number of pixels across one dimension of a basic block size unit; and

storing data of the picture in basic block size units of pixels mapped in an array therein with data across each scan line of the picture being stored in sequence in a single row of the memory, with data of pixels of the same basic block size unit that appear vertically adjacent in the picture being stored in a common column in adjacent rows of the memory.

3. A method of mapping in a buffer memory a digital video data picture composed of a plurality of pixels, rectangular macroblocks of the pixels and horizontal picture-wide scan lines of the pixels, the method comprising the steps of:

providing a buffer memory physically divided into a plurality of consecutively addressable pages, each page having consecutively addressable storage locations of a plurality sufficient to store data for all pixels of at least a macroblock of the picture;

storing the picture in the buffer memory with data for each pixel of at least one macroblock being stored on a single page, with data of at least one column of vertically adjacent line segments of one or more pixels each being stored consecutively on the page beginning at locations spaced a first fixed increment apart and with the data of each horizontally adjacent column being stored consecutively in the memory beginning at locations space a second fixed increment apart.

4. The method claim 3 wherein:

each line segment includes a plurality of horizontally adjacent pixels; and

the storing step includes the step of storing the data of the pixels of line segments within a macroblock in consecutive storage locations.

5. The method claim 3 wherein:

each line segment includes a plurality of horizontally adjacent pixels; and  
the storing step includes the step of storing the data of the pixels of an adjacent plurality of line segments in consecutive storage locations.

5 6. The method claim 3 wherein:

the picture is a full frame picture having interleaved odd and even scan lines; and  
the storing step includes the step of storing the picture in the buffer memory with data of at least one column of vertically adjacent line segments alternating odd and even line segments consecutively on the page beginning at locations spaced the first fixed increment apart.

10 7. The method claim 3 wherein:

the picture is a field picture containing either odd and even scan lines of a full frame picture; and  
the storing step includes the step of storing the field picture in the buffer memory with data of at least one column of vertically adjacent line segments of either odd and even lines of the full frame picture.

8. The method claim 3 further comprising the step of:

15 storing in a page of the memory data of a rectangular array of pixels of the picture.

9. The method claim 3 further comprising the step of:

consecutively reading line segments of data of a scan line of the picture by addressing the memory sequentially by the second fixed increment.

10. The method claim 3 further comprising the step of:

20 consecutively reading data of a rectangular array of pixels of at least one column of the picture by addressing the memory sequentially by the first fixed increment.

11. The method claim 3 further comprising the step of:

25 consecutively reading data of a rectangular array of pixels of more than one column of the picture by addressing columns of data by the second fixed increment and the line segments of each column also sequentially by the first fixed increment.

12. The method claim 3 wherein:

each line segment includes a plurality of eight horizontally adjacent pixels; and  
the storing step includes the step of storing the data of the pixels of line segments within a macroblock in eight consecutive storage locations.



- 27 -

13. The method claim 3 wherein:

the storing step includes the step of storing the data of the pixels of an adjacent plurality of 16 line segments in consecutive storage locations.

14. The method claim 3 further comprising the step of:

5 storing in a page of the memory data of a rectangular array of 32x16 pixels of the picture.

15. The method claim 3 further comprising the step of:

consecutively reading line segments of data of a scan line of the picture by addressing the memory sequentially by the fixed increment 80 hex.

16. The method claim 3 further comprising the step of:

10 consecutively reading data of a rectangular array of pixels of at least one column of the picture by addressing the memory sequentially by 8 hex.

17. The method claim 3 further comprising the step of:

consecutively reading data of a rectangular array of pixels of more than one column of the picture by addressing the data by increments of 8 hex within increments of 80 hex.

15 18. The method claim 3 wherein:

the storing step includes the step of storing the picture in the buffer memory with data for each of a plurality of macroblocks arranged so that no rectangular macroblock size array of pixel data is stored on more than four pages of the buffer memory.

19. The method claim 3 wherein:

20 the providing step includes the step of providing a buffer memory divided into a plurality of consecutively addressable pages, each page having consecutively addressable storage locations of a plurality sufficient to store data for all pixels of at least two macroblocks of the picture; and

the storing step includes the step of storing the picture in the buffer memory with data for each of a plurality of macroblocks arranged so that no rectangular macroblock size array of pixel data is stored on more  
25 than three pages of the buffer memory.

20. The method claim 3 wherein:

the providing step includes the step of providing a buffer memory divided into a plurality of consecutively addressable pages, each page having consecutively addressable storage locations of a plurality sufficient to store data for all pixels of at least two macroblocks of the picture; and

30 the storing step includes the step of storing in the buffer memory all column-wide series of data for any macroblock sized array of pixels beginning at any half pel location of the picture on no more than three pages of the buffer memory.

21. A motion compensation circuit for use in a video decoder that reconstructs a current picture based upon a motion vector in an input video stream, the motion compensation circuit comprising:

a memory including rows and columns of memory locations, at least a portion of the memory locations collectively holding reference pictures, wherein the reference pictures include a plurality of reference macroblocks;

means for extracting the motion vector from the input video stream, the motion vector pointing to a sub-portion of the memory locations that collectively hold one of said reference pictures;

motion compensation address and block size generation means, responsive to the motion extracted motion vector, that generates a plurality of address and block size pairs such that a sub-sub-portion of the memory defined by each address and block size pair does not include a row crossing; and

means for reading the reference pictures from the memory in a plurality of parts and responsive to the address and block size pairs.

22. A method of addressing a rectangular array of digital picture data stored in a buffer memory, the method comprising the steps of:

processing picture data with a first circuit to determine whether the rectangular array of data crosses a page boundary that results in the array lying in non-contiguous portions of the memory, and if so, dividing the addressing process into a plurality of addressing operations, one for each rectangular portion of the array lying in each of the non-contiguous portions of the memory; then

for each addressing operation, processing data with a memory controller circuit to determine whether the respective rectangular portion of the array lies on a plurality of contiguous pages, and if so, dividing the respective addressing operation into a plurality of addressing steps, one for each rectangular sub-portion of the respective rectangular portion of array lying in each of the contiguous pages of the memory; then

addressing the rectangular array by sequentially addressing each sub-portion of each portion of the array.

23. The method of claim 22 wherein:

the step of processing picture data with the first circuit includes the step of processing motion vector data with a motion compensation circuit to locate a rectangular array of data in the memory and read from the rectangular array of data from the memory by determining page boundary crossings that result in the array lying in non-adjacent pages of the memory, and if so, dividing the addressing process into the plurality of addressing operations with portions of the array lying on adjacent pages of the memory being included in the same addressing process.

1/19

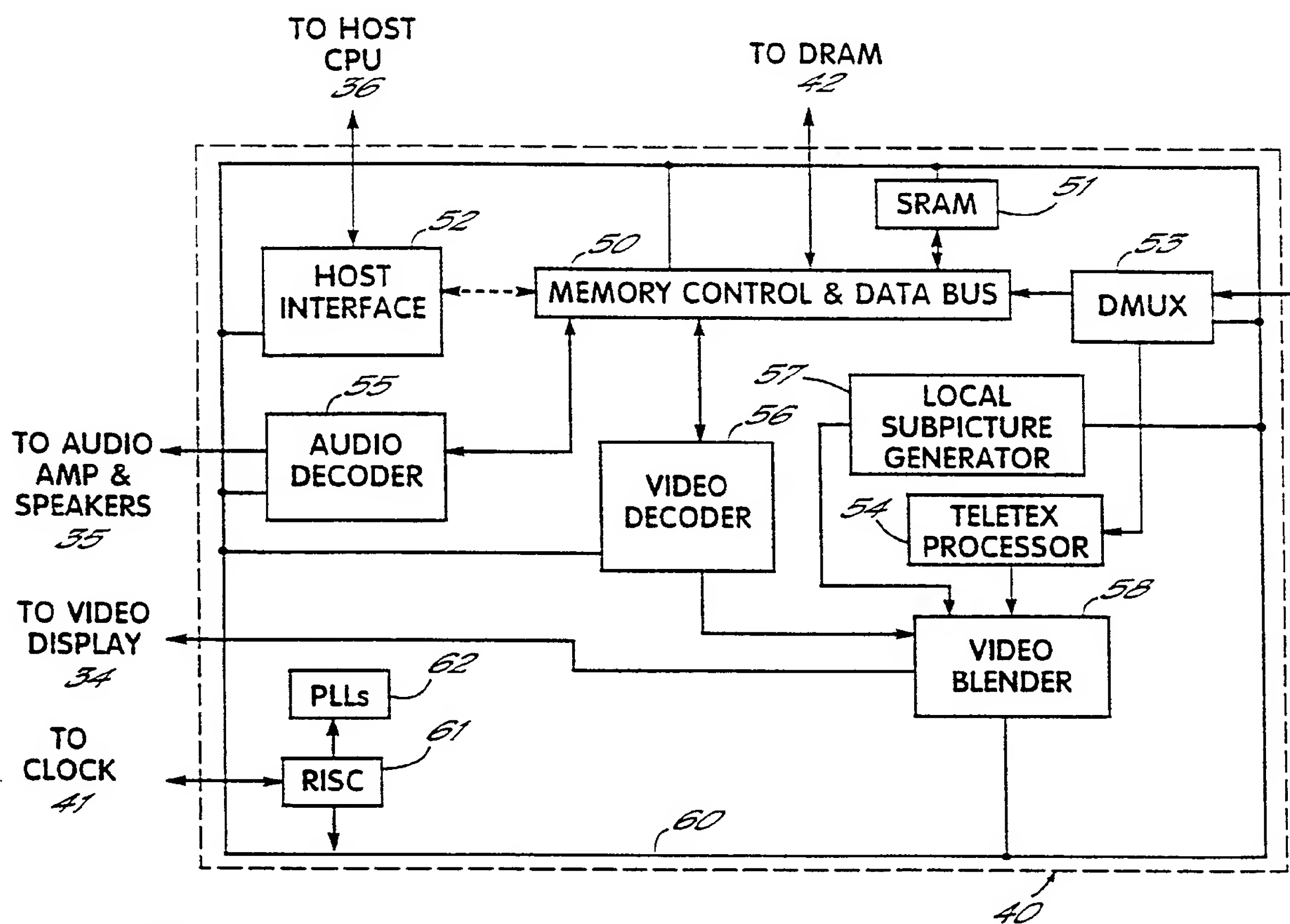
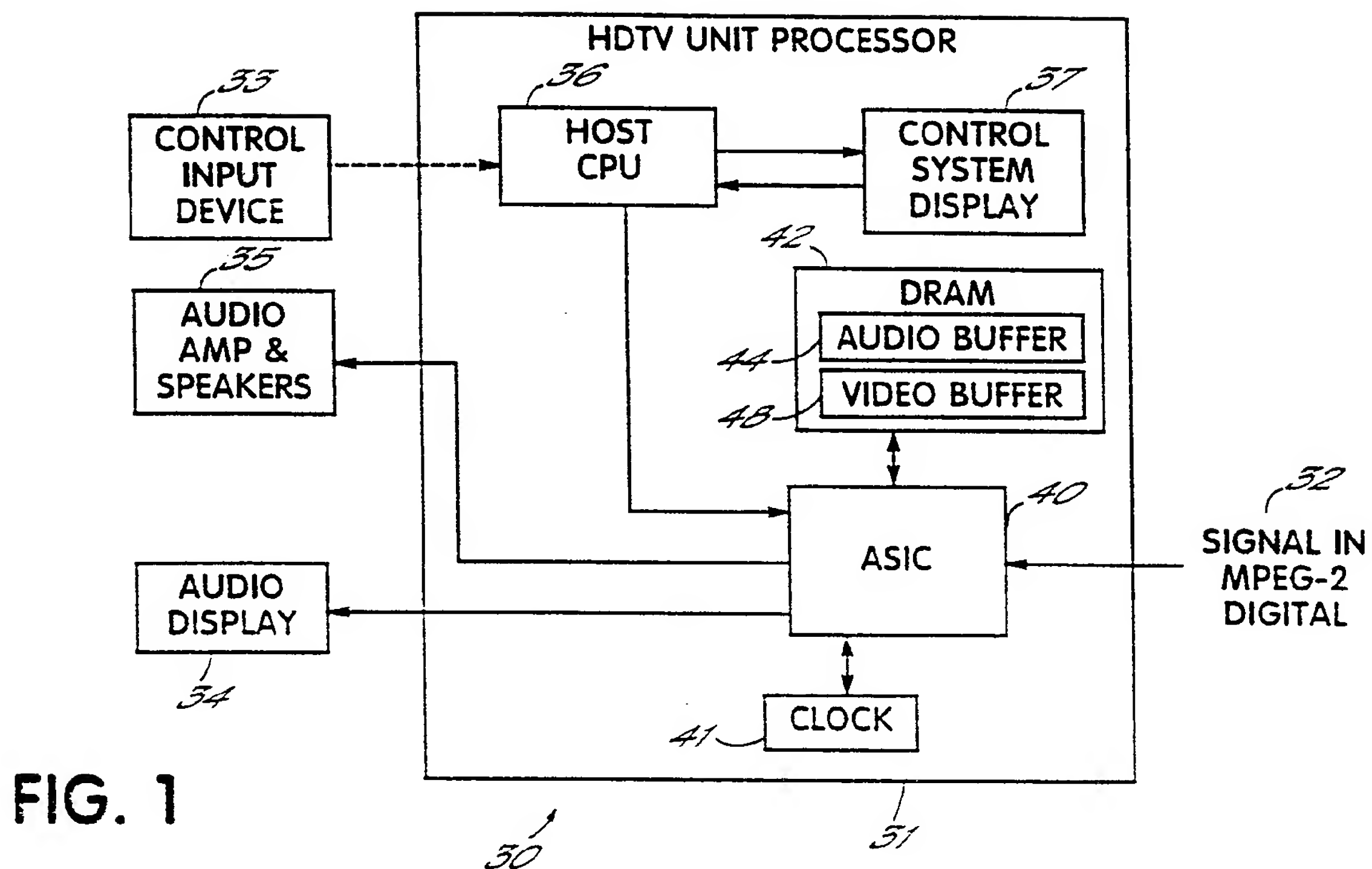


FIG. 3

Y <sub>0,0</sub>	Y <sub>0,1</sub>	Y <sub>0,2</sub>	Y <sub>0,3</sub>	Y <sub>0,4</sub>	Y <sub>0,5</sub>	Y <sub>0,6</sub>	Y <sub>0,7</sub>
C <sub>0,0</sub>		C <sub>0,2</sub>		C <sub>0,4</sub>		C <sub>0,6</sub>	
Y <sub>1,0</sub>	Y <sub>1,1</sub>	Y <sub>1,2</sub>	Y <sub>1,3</sub>	Y <sub>1,4</sub>	Y <sub>1,5</sub>	Y <sub>1,6</sub>	Y <sub>1,7</sub>
Y <sub>2,0</sub>	Y <sub>2,1</sub>	Y <sub>2,2</sub>	Y <sub>2,3</sub>	Y <sub>2,4</sub>	Y <sub>2,5</sub>	Y <sub>2,6</sub>	Y <sub>2,7</sub>
C <sub>2,0</sub>		C <sub>2,2</sub>		C <sub>2,4</sub>		C <sub>2,6</sub>	
Y <sub>3,0</sub>	Y <sub>3,1</sub>	Y <sub>3,2</sub>	Y <sub>3,3</sub>	Y <sub>3,4</sub>	Y <sub>3,5</sub>	Y <sub>3,6</sub>	Y <sub>3,7</sub>
Y <sub>4,0</sub>	Y <sub>4,1</sub>	Y <sub>4,2</sub>	Y <sub>4,3</sub>	Y <sub>4,4</sub>	Y <sub>4,5</sub>	Y <sub>4,6</sub>	Y <sub>4,7</sub>
C <sub>4,0</sub>		C <sub>4,2</sub>		C <sub>4,4</sub>		C <sub>4,6</sub>	
Y <sub>5,0</sub>	Y <sub>5,1</sub>	Y <sub>5,2</sub>	Y <sub>5,3</sub>	Y <sub>5,4</sub>	Y <sub>5,5</sub>	Y <sub>5,6</sub>	Y <sub>5,7</sub>
Y <sub>6,0</sub>	Y <sub>6,1</sub>	Y <sub>6,2</sub>	Y <sub>6,3</sub>	Y <sub>6,4</sub>	Y <sub>6,5</sub>	Y <sub>6,6</sub>	Y <sub>6,7</sub>
C <sub>6,0</sub>		C <sub>6,2</sub>		C <sub>6,4</sub>		C <sub>6,6</sub>	
Y <sub>7,0</sub>	Y <sub>7,1</sub>	Y <sub>7,2</sub>	Y <sub>7,3</sub>	Y <sub>7,4</sub>	Y <sub>7,5</sub>	Y <sub>7,6</sub>	Y <sub>7,7</sub>

FIG. 4

ODD FIELD							
Y <sub>0,0</sub>	Y <sub>0,1</sub>	Y <sub>0,2</sub>	Y <sub>0,3</sub>	Y <sub>0,4</sub>	Y <sub>0,5</sub>	Y <sub>0,6</sub>	Y <sub>0,7</sub>
C <sub>0,0</sub>		C <sub>0,2</sub>		C <sub>0,4</sub>		C <sub>0,6</sub>	
Y <sub>2,0</sub>	Y <sub>2,1</sub>	Y <sub>2,2</sub>	Y <sub>2,3</sub>	Y <sub>2,4</sub>	Y <sub>2,5</sub>	Y <sub>2,6</sub>	Y <sub>2,7</sub>
Y <sub>4,0</sub>	Y <sub>4,1</sub>	Y <sub>4,2</sub>	Y <sub>4,3</sub>	Y <sub>4,4</sub>	Y <sub>4,5</sub>	Y <sub>4,6</sub>	Y <sub>4,7</sub>
C <sub>4,0</sub>		C <sub>4,2</sub>		C <sub>4,4</sub>		C <sub>4,6</sub>	
Y <sub>6,0</sub>	Y <sub>6,1</sub>	Y <sub>6,2</sub>	Y <sub>6,3</sub>	Y <sub>6,4</sub>	Y <sub>6,5</sub>	Y <sub>6,6</sub>	Y <sub>6,7</sub>
EVEN FIELD							
Y <sub>1,0</sub>	Y <sub>1,1</sub>	Y <sub>1,2</sub>	Y <sub>1,3</sub>	Y <sub>1,4</sub>	Y <sub>1,5</sub>	Y <sub>1,6</sub>	Y <sub>1,7</sub>
C <sub>2,0</sub>		C <sub>2,2</sub>		C <sub>2,4</sub>		C <sub>2,6</sub>	
Y <sub>3,0</sub>	Y <sub>3,1</sub>	Y <sub>3,2</sub>	Y <sub>3,3</sub>	Y <sub>3,4</sub>	Y <sub>3,5</sub>	Y <sub>3,6</sub>	Y <sub>3,7</sub>
Y <sub>5,0</sub>	Y <sub>5,1</sub>	Y <sub>5,2</sub>	Y <sub>5,3</sub>	Y <sub>5,4</sub>	Y <sub>5,5</sub>	Y <sub>5,6</sub>	Y <sub>5,7</sub>
C <sub>6,0</sub>		C <sub>6,2</sub>		C <sub>6,4</sub>		C <sub>6,6</sub>	
Y <sub>7,0</sub>	Y <sub>7,1</sub>	Y <sub>7,2</sub>	Y <sub>7,3</sub>	Y <sub>7,4</sub>	Y <sub>7,5</sub>	Y <sub>7,6</sub>	Y <sub>7,7</sub>



Y <sub>0,0</sub>	Y <sub>0,1</sub>	Y <sub>0,2</sub>	Y <sub>0,3</sub>	Y <sub>0,4</sub>	Y <sub>0,5</sub>	Y <sub>0,6</sub>	Y <sub>0,7</sub>
C <sub>0,0</sub>		C <sub>0,2</sub>		C <sub>0,4</sub>		C <sub>0,6</sub>	
Y <sub>1,0</sub>	Y <sub>1,1</sub>	Y <sub>1,2</sub>	Y <sub>1,3</sub>	Y <sub>1,4</sub>	Y <sub>1,5</sub>	Y <sub>1,6</sub>	Y <sub>1,7</sub>
C <sub>1,0</sub>		C <sub>1,2</sub>		C <sub>1,4</sub>		C <sub>1,6</sub>	
Y <sub>2,0</sub>	Y <sub>2,1</sub>	Y <sub>2,2</sub>	Y <sub>2,3</sub>	Y <sub>2,4</sub>	Y <sub>2,5</sub>	Y <sub>2,6</sub>	Y <sub>2,7</sub>
C <sub>2,0</sub>		C <sub>2,2</sub>		C <sub>2,4</sub>		C <sub>2,6</sub>	
Y <sub>3,0</sub>	Y <sub>3,1</sub>	Y <sub>3,2</sub>	Y <sub>3,3</sub>	Y <sub>3,4</sub>	Y <sub>3,5</sub>	Y <sub>3,6</sub>	Y <sub>3,7</sub>
C <sub>3,0</sub>		C <sub>3,2</sub>		C <sub>3,4</sub>		C <sub>3,6</sub>	
Y <sub>4,0</sub>	Y <sub>4,1</sub>	Y <sub>4,2</sub>	Y <sub>4,3</sub>	Y <sub>4,4</sub>	Y <sub>4,5</sub>	Y <sub>4,6</sub>	Y <sub>4,7</sub>
C <sub>4,0</sub>		C <sub>4,2</sub>		C <sub>4,4</sub>		C <sub>4,6</sub>	
Y <sub>5,0</sub>	Y <sub>5,1</sub>	Y <sub>5,2</sub>	Y <sub>5,3</sub>	Y <sub>5,4</sub>	Y <sub>5,5</sub>	Y <sub>5,6</sub>	Y <sub>5,7</sub>
C <sub>5,0</sub>		C <sub>5,2</sub>		C <sub>5,4</sub>		C <sub>5,6</sub>	
Y <sub>6,0</sub>	Y <sub>6,1</sub>	Y <sub>6,2</sub>	Y <sub>6,3</sub>	Y <sub>6,4</sub>	Y <sub>6,5</sub>	Y <sub>6,6</sub>	Y <sub>6,7</sub>
C <sub>6,0</sub>		C <sub>6,2</sub>		C <sub>6,4</sub>		C <sub>6,6</sub>	
Y <sub>7,0</sub>	Y <sub>7,1</sub>	Y <sub>7,2</sub>	Y <sub>7,3</sub>	Y <sub>7,4</sub>	Y <sub>7,5</sub>	Y <sub>7,6</sub>	Y <sub>7,7</sub>
C <sub>7,0</sub>		C <sub>7,2</sub>		C <sub>7,4</sub>		C <sub>7,6</sub>	

FIG. 3A

EVEN FIELD							
Y <sub>0,0</sub>	Y <sub>0,1</sub>	Y <sub>0,2</sub>	Y <sub>0,3</sub>	Y <sub>0,4</sub>	Y <sub>0,5</sub>	Y <sub>0,6</sub>	Y <sub>0,7</sub>
C <sub>0,0</sub>		C <sub>0,2</sub>		C <sub>0,4</sub>		C <sub>0,6</sub>	
Y <sub>2,0</sub>	Y <sub>2,1</sub>	Y <sub>2,2</sub>	Y <sub>2,3</sub>	Y <sub>2,4</sub>	Y <sub>2,5</sub>	Y <sub>2,6</sub>	Y <sub>2,7</sub>
C <sub>2,0</sub>		C <sub>2,2</sub>		C <sub>2,4</sub>		C <sub>2,6</sub>	
Y <sub>4,0</sub>	Y <sub>4,1</sub>	Y <sub>4,2</sub>	Y <sub>4,3</sub>	Y <sub>4,4</sub>	Y <sub>4,5</sub>	Y <sub>4,6</sub>	Y <sub>4,7</sub>
C <sub>4,0</sub>		C <sub>4,2</sub>		C <sub>4,4</sub>		C <sub>4,6</sub>	
Y <sub>6,0</sub>	Y <sub>6,1</sub>	Y <sub>6,2</sub>	Y <sub>6,3</sub>	Y <sub>6,4</sub>	Y <sub>6,5</sub>	Y <sub>6,6</sub>	Y <sub>6,7</sub>
C <sub>6,0</sub>		C <sub>6,2</sub>		C <sub>6,4</sub>		C <sub>6,6</sub>	
ODD FIELD							
Y <sub>1,0</sub>	Y <sub>1,1</sub>	Y <sub>1,2</sub>	Y <sub>1,3</sub>	Y <sub>1,4</sub>	Y <sub>1,5</sub>	Y <sub>1,6</sub>	Y <sub>1,7</sub>
C <sub>1,0</sub>		C <sub>1,2</sub>		C <sub>1,4</sub>		C <sub>1,6</sub>	
Y <sub>3,0</sub>	Y <sub>3,1</sub>	Y <sub>3,2</sub>	Y <sub>3,3</sub>	Y <sub>3,4</sub>	Y <sub>3,5</sub>	Y <sub>3,6</sub>	Y <sub>3,7</sub>
C <sub>3,0</sub>		C <sub>3,2</sub>		C <sub>3,4</sub>		C <sub>3,6</sub>	
Y <sub>5,0</sub>	Y <sub>5,1</sub>	Y <sub>5,2</sub>	Y <sub>5,3</sub>	Y <sub>5,4</sub>	Y <sub>5,5</sub>	Y <sub>5,6</sub>	Y <sub>5,7</sub>
C <sub>5,0</sub>		C <sub>5,2</sub>		C <sub>5,4</sub>		C <sub>5,6</sub>	
Y <sub>7,0</sub>	Y <sub>7,1</sub>	Y <sub>7,2</sub>	Y <sub>7,3</sub>	Y <sub>7,4</sub>	Y <sub>7,5</sub>	Y <sub>7,6</sub>	Y <sub>7,7</sub>
C <sub>7,0</sub>		C <sub>7,2</sub>		C <sub>7,4</sub>		C <sub>7,6</sub>	

FIG. 5

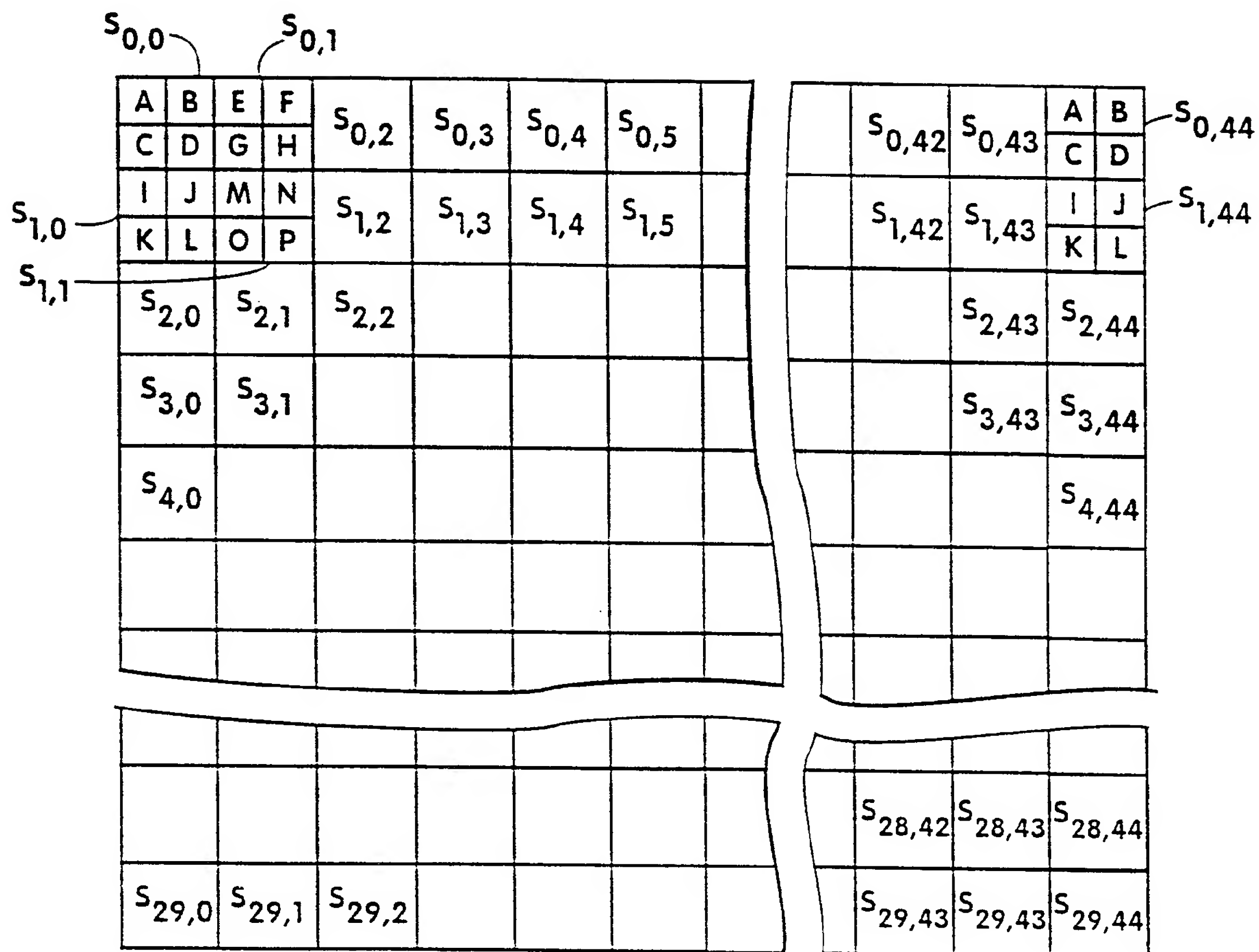


FIG. 6

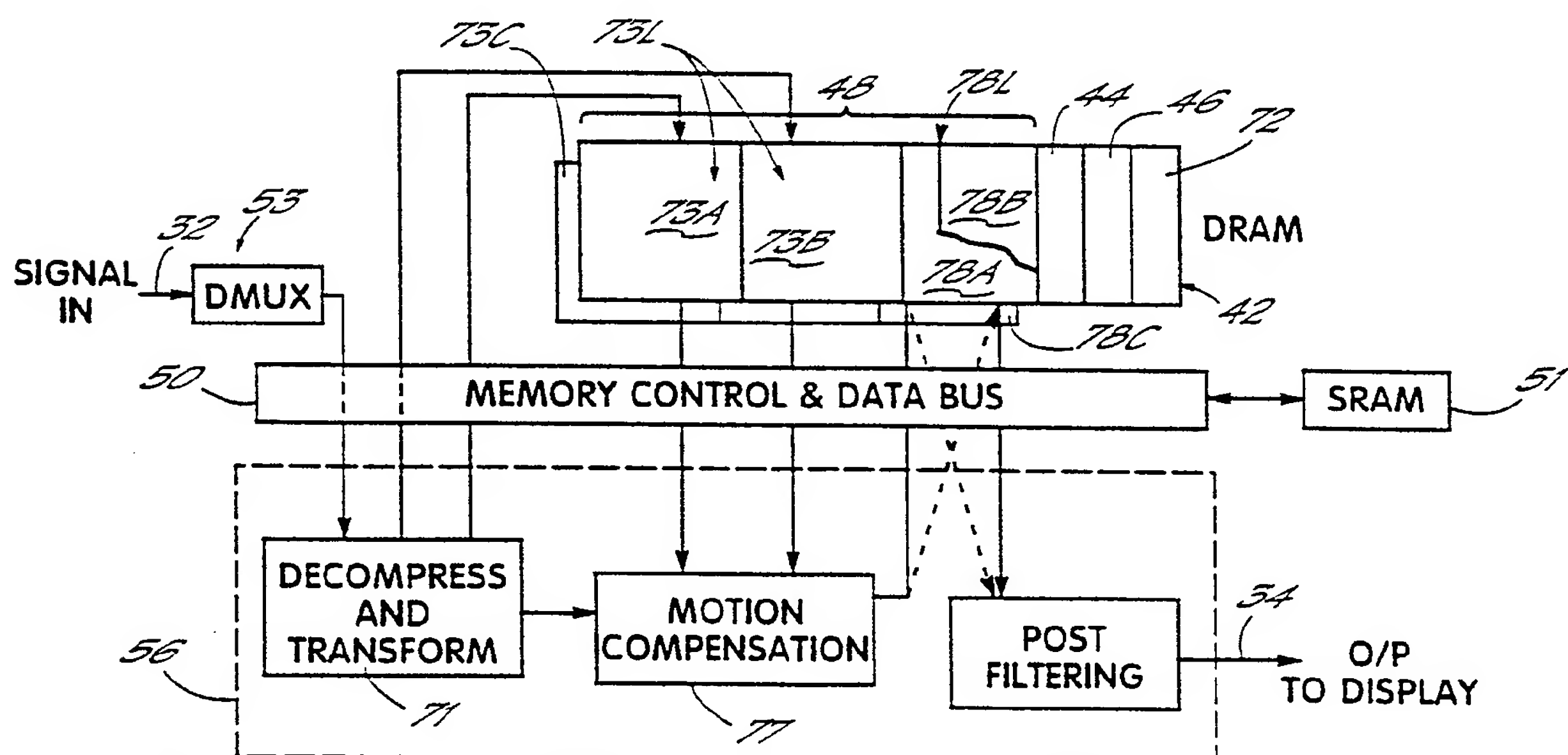


FIG. 7

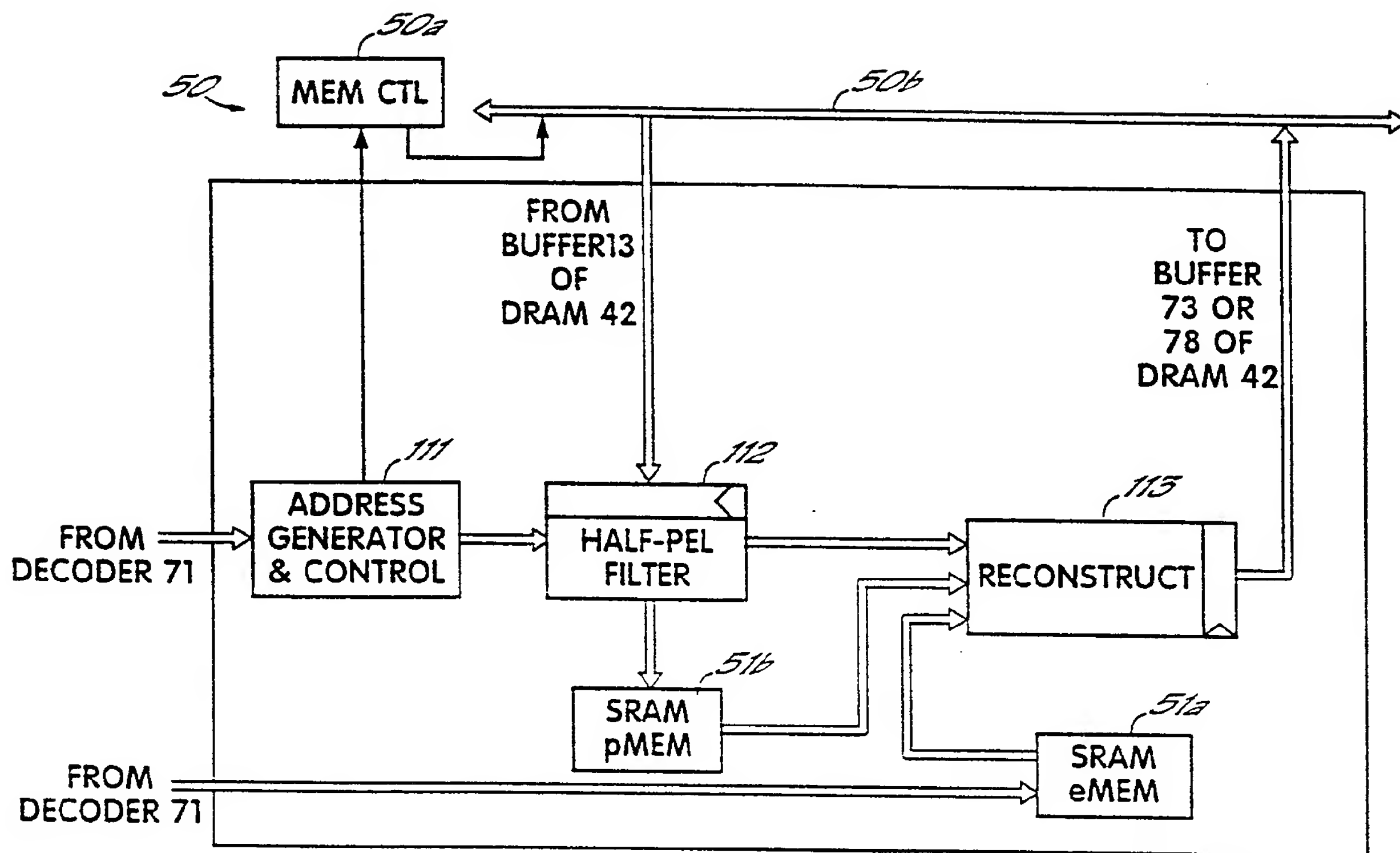


FIG. 7A

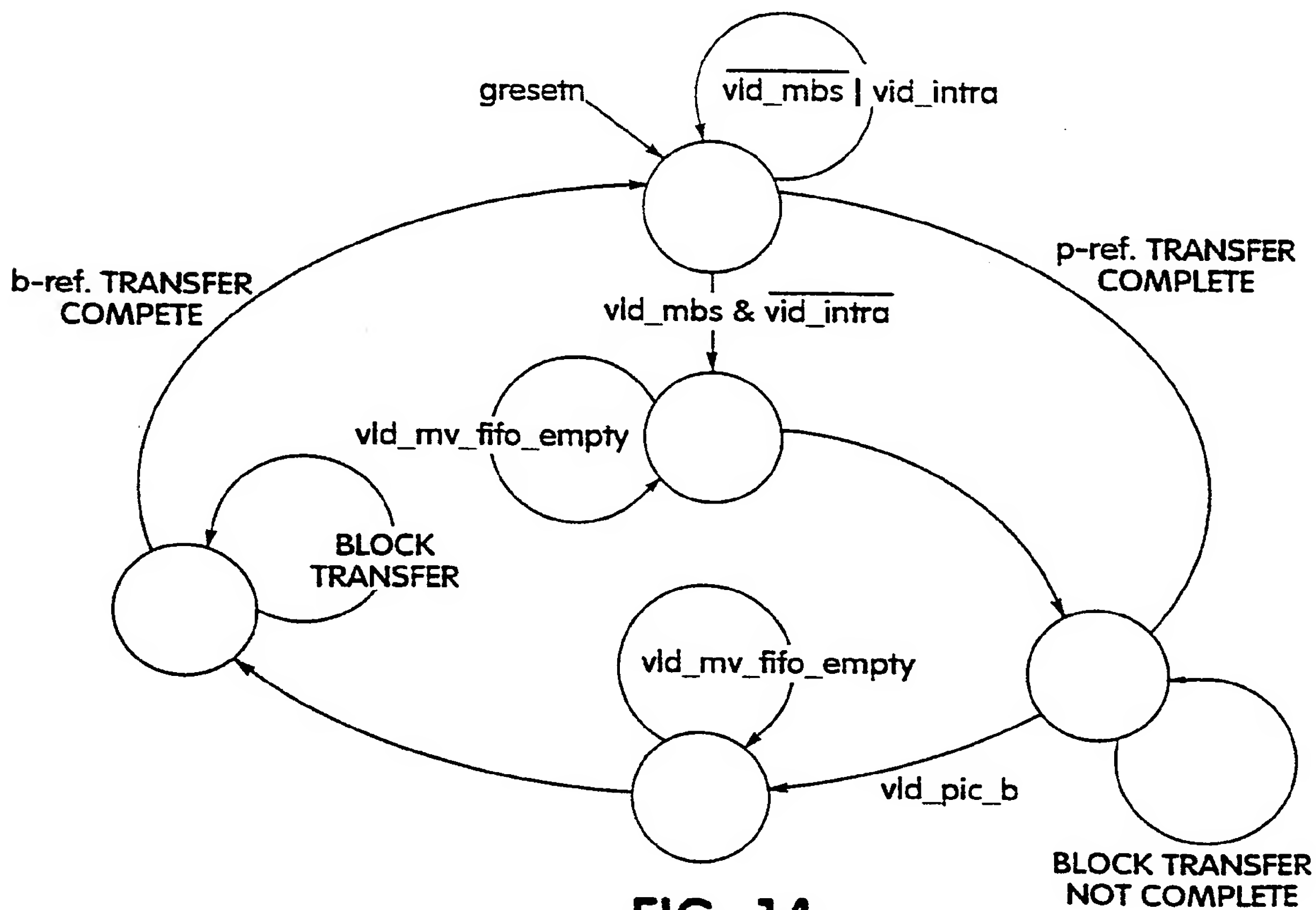


FIG. 14





B FRAME, LUMINANCE,  
EVEN FIELD

76	04	81		81		80h		160h										
74-0	Y00	Y01	Y02	Y03	Y04	Y05	Y06	Y07	Y00	Y01	Y02	Y03	Y04	Y05	Y06	Y07	Y00	....
74-1	Y20	Y21	Y22	Y	A		4	Y25	Y26	Y27	B		4	Y25	Y26	Y27	....	E
74-2	Y40	Y41	Y42	Y	A		4	Y45	Y46	Y47	B		4	Y45	Y46	Y47		
	Y60	Y61	Y62	Y63	164	Y65	Y66	Y	50	Y61	Y62	Y63	164	Y65	Y66	Y67		50
	Y00	Y01	Y02	Y03	Y04	Y05	Y06		50	Y01	Y02	Y03	Y04	Y05	Y06	Y07	Y00	....
	Y20	Y21	Y22	Y	C		4	Y25	Y26	Y27	D		4	Y25	Y26	Y27	....	G
	Y40	Y41	Y42	Y	C		4	Y45	Y46	Y47	D		4	Y45	Y46	Y47		
	Y60	Y61	Y62	Y63	164	Y65	Y66	Y67		Y61	Y62	Y63	164	Y65	Y66	Y67		
	Y00	Y01	Y02	Y03	Y04	Y05	Y06	Y07		Y01	Y02	Y03	Y04	Y05	Y06	Y07	Y00	....
	Y20	Y21	Y22	Y	I		4	Y25	Y26	Y27	J		4	Y25	Y26	Y27	....	M
	Y40	Y41	Y42	Y	I		4	Y45	Y46	Y47	J		4	Y45	Y46	Y47		
	Y60	Y61	Y62	Y63	164	Y65	Y66	Y	50	Y61	Y62	Y63	164	Y65	Y66	Y67		50
	Y00	Y01	Y02	Y03	Y04	Y05	Y06	Y	50	Y01	Y02	Y03	Y04	Y05	Y06	Y07	Y00	....
	Y20	Y21	Y22	Y	K		4	Y25	Y26	Y27	L		4	Y25	Y26	Y27	....	O
	Y40	Y41	Y42	Y	K		4	Y45	Y46	Y47	L		4	Y45	Y46	Y47		
74-15	Y60	Y61	Y62	Y63	164	Y65	Y66	Y67		Y61	Y62	Y63	164	Y65	Y66	Y67		

FIG. 8A

4:2:2 CHROMINANCE,  
I OR PFRAME

73C

76

OH

80H

160H

U00	V00	U02	V02	U04	V04	U06	V06	U00	V00	U02	V02	U04	V04	U06	V06	U00	.....
U10	V10	U12	V12	U14	V14	U16	V16	U10	V10	U12	V12	U14	V14	U16	V16	.....	
U20	V20	U22	V22	U24	V24	U26	V26	U20	V20	U22	V22	U24	V24	U26	V26		
U30	V30	U32	A	14V34	U36	V36		U30	V30	U32	B	134V34	U36	V36			E
U40	V40	U42		14V44	U46	V46		U40	V40	U42		144V44	U46	V46			
U50	V50	U52	V52	U54	V54	U56	V56	U50	V50	U52	V52	U54	V54	U56	V56		
U60	V60	U62	V62	U64	V64	U66	V66	U60	V60	U62	V62	U64	V64	U66	V66		
U70	V70	U72	V72	U74	V74	U76	V76	U70	V70	U72	V72	U74	V74	U76	V76		
U00	V00	U02	V02	U04	V04	U06	V06	U00	V00	U02	V02	U04	V04	U06	V06	U00	.....
U10	V10	U12	V12	U14	V14	U16	V16	U10	V10	U12	V12	U14	V14	U16	V16	.....	
U20	V20	U22	V22	U24	V24	U26	V26	U20	V20	U22	V22	U24	V24	U26	V26		
U30	V30	U32	C	14V34	U36	V36		U30	V30	U32	D	134V34	U36	V36			G
U40	V40	U42		14V44	U46	V46		U40	V40	U42		144V44	U46	V46			
U50	V50	U52	V52	U54	V54	U56	V56	U50	V50	U52	V52	U54	V54	U56	V56		
U60	V60	U62	V62	U64	V64	U66	V66	U60	V60	U62	V62	U64	V64	U66	V66		
U70	V70	U72	V72	U74	V74	U76	V76	U70	V70	U72	V72	U74	V74	U76	V76		

FIG. 9

4:2:2 CHROMINANCE, B FRAME,  
EVEN FIELD

76

81

U00	V00	U02	V02	U04	V04	U06	V06	U00	V00	U02	V02	U04	V04	U06	V06		
U20	V20	U22	V22	A	V24	U26	V26	U20	V20	U22	V22	B	V24	U26	V26		A
U40	V40	U42	V42		V44	U46	V46	U40	V40	U42	V42		V44	U46	V46		
U60	V60	U62	V62	U64	V64	U66	V66	U60	V60	U62	V62	U64	V64	U66	V66		
U00	V00	U02	V02	U04	V04	U06	V06	U00	V00	U02	V02	U04	V04	U06	V06		
U20	V20	U22	V22	C	V24	U26	V26	U20	V20	U22	V22	D	V24	U26	V26		C
U40	V40	U42	V42		V44	U46	V46	U40	V40	U42	V42		V44	U46	V46		
U60	V60	U62	V62	U64	V64	U66	V66	U60	V60	U62	V62	U64	V64	U66	V66		
U00	V00	U02	V02	U04	V04	U06	V06	U00	V00	U02	V02	U04	V04	U06	V06		
U20	V20	U22	V22	I	V24	U26	V26	U20	V20	U22	V22	J	V24	U26	V26		I
U40	V40	U42	V42		V44	U46	V46	U40	V40	U42	V42		V44	U46	V46		
U60	V60	U62	V62	U64	V64	U66	V66	U60	V60	U62	V62	U64	V64	U66	V66		
U00	V00	U02	V02	U04	V04	U06	V06	U00	V00	U02	V02	U04	V04	U06	V06		
U20	V20	U22	V22	K	V24	U26	V26	U20	V20	U22	V22	L	V24	U26	V26		K
U40	V40	U42	V42		V44	U46	V46	U40	V40	U42	V42		V44	U46	V46		
U60	V60	U62	V62	U64	V64	U66	V66	U60	V60	U62	V62	U64	V64	U66	V66		

FIG. 9A



4:2:0 CHROMINANCE,  
I OR P FRAME

76 81 81

U00	V00	U02	V02	U04	V04	U06	V06	U00	V00	U02	V02	U04	V04	U06	V06		
U20	V20	U22	A	'4	V24	U26	V26	U20	V20	U22	B	'24	V24	U26	V26		A
U40	V40	U42		'4	V44	U46	V46	U40	V40	U42		44	V44	U46	V46		
U60	V60	U62	V62	U64	V64	U66	V66	U60	V60	U62	V62	U64	V64	U66	V66		
U00	V00	U02	V02	U04	V04	U06	V06	U00	V00	U02	V02	U04	V04	U06	V06		
U20	V20	U22	C	'4	V24	U26	V26	U20	V20	U22	D	'24	V24	U26	V26		C
U40	V40	U42		'4	V44	U46	V46	U40	V40	U42		44	V44	U46	V46		
U60	V60	U62	V62	U64	V64	U66	V66	U60	V60	U62	V62	U64	V64	U66	V66		
U00	V00	U02	V02	U04	V04	U06	V06	U00	V00	U02	V02	U04	V04	U06	V06		
U20	V20	U22	I	'4	V24	U26	V26	U20	V20	U22	J	'24	V24	U26	V26		I
U40	V40	U42		'4	V44	U46	V46	U40	V40	U42		44	V44	U46	V46		
U60	V60	U62	V62	U64	V64	U66	V66	U60	V60	U62	V62	U64	V64	U66	V66		
U00	V00	U02	V02	U04	V04	U06	V06	U00	V00	U02	V02	U04	V04	U06	V06		
U20	V20	U22	K	'4	V24	U26	V26	U20	V20	U22	L	'24	V24	U26	V26		K
U40	V40	U42		'4	V44	U46	V46	U40	V40	U42		44	V44	U46	V46		
U60	V60	U62	V62	U64	V64	U66	V66	U60	V60	U62	V62	U64	V64	U66	V66		

FIG. 10

4:2:0 CHROMINANCE, B FRAME,  
EVEN FIELD

804

U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	A
U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	
U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	C
U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	
U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	I
U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	
U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	K
U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	
U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	A
U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	
U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	C
U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	
U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	I
U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	
U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	U00 V00 U02 V	$\frac{1}{4}V_{04}U_{06}V_{06}$	K
U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	U40 V40 U42 V	$\frac{4}{4}V_{44}U_{46}V_{46}$	

FIG. 10 A

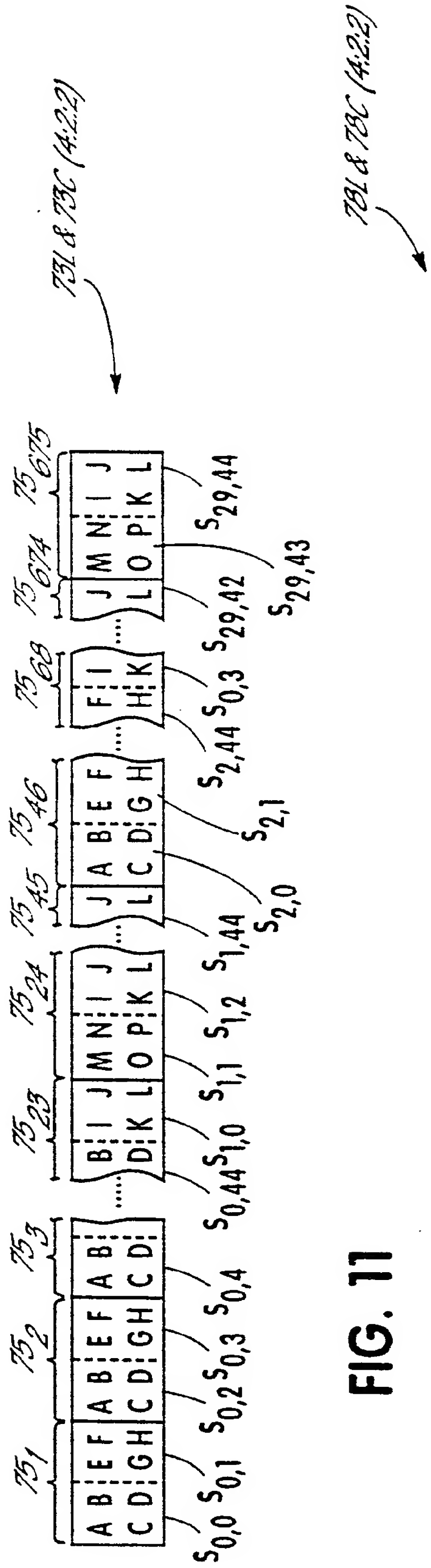


FIG. 11

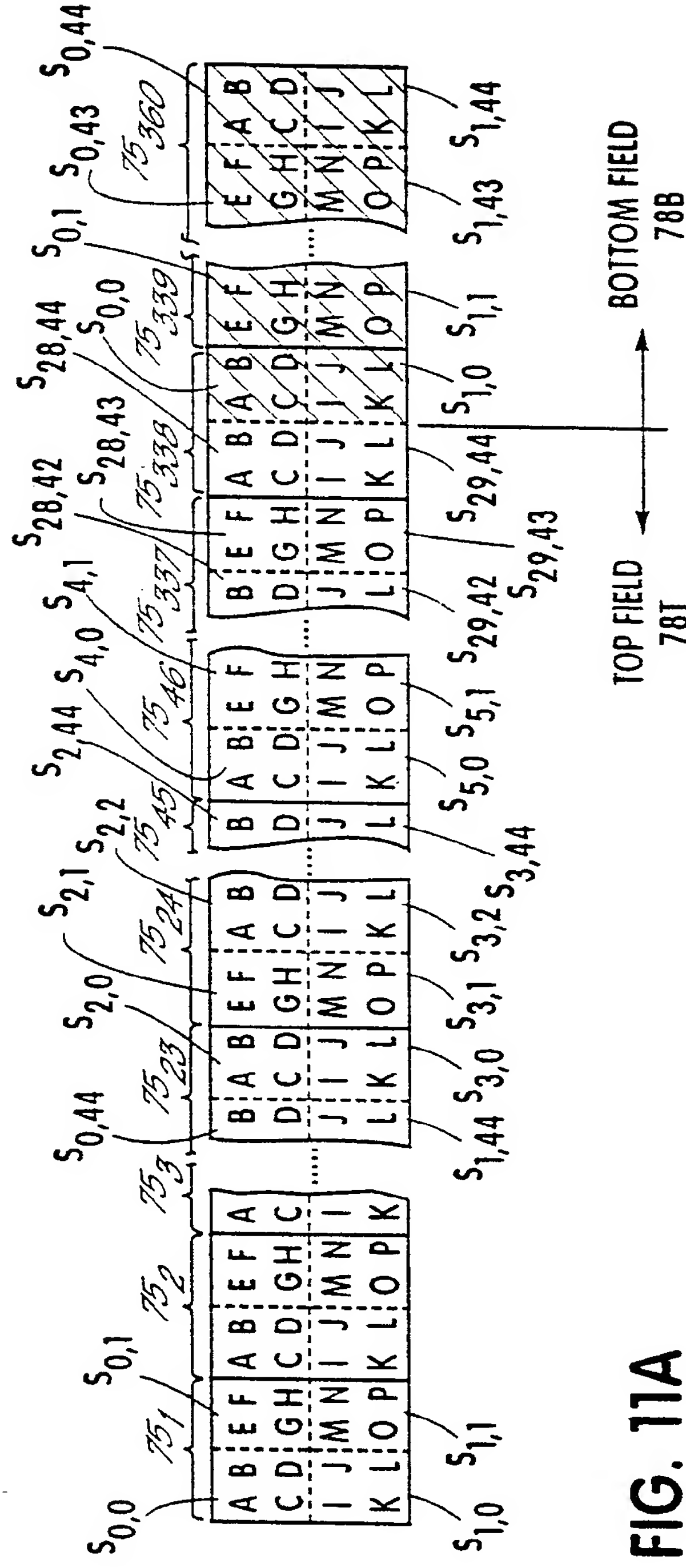


FIG. 11A

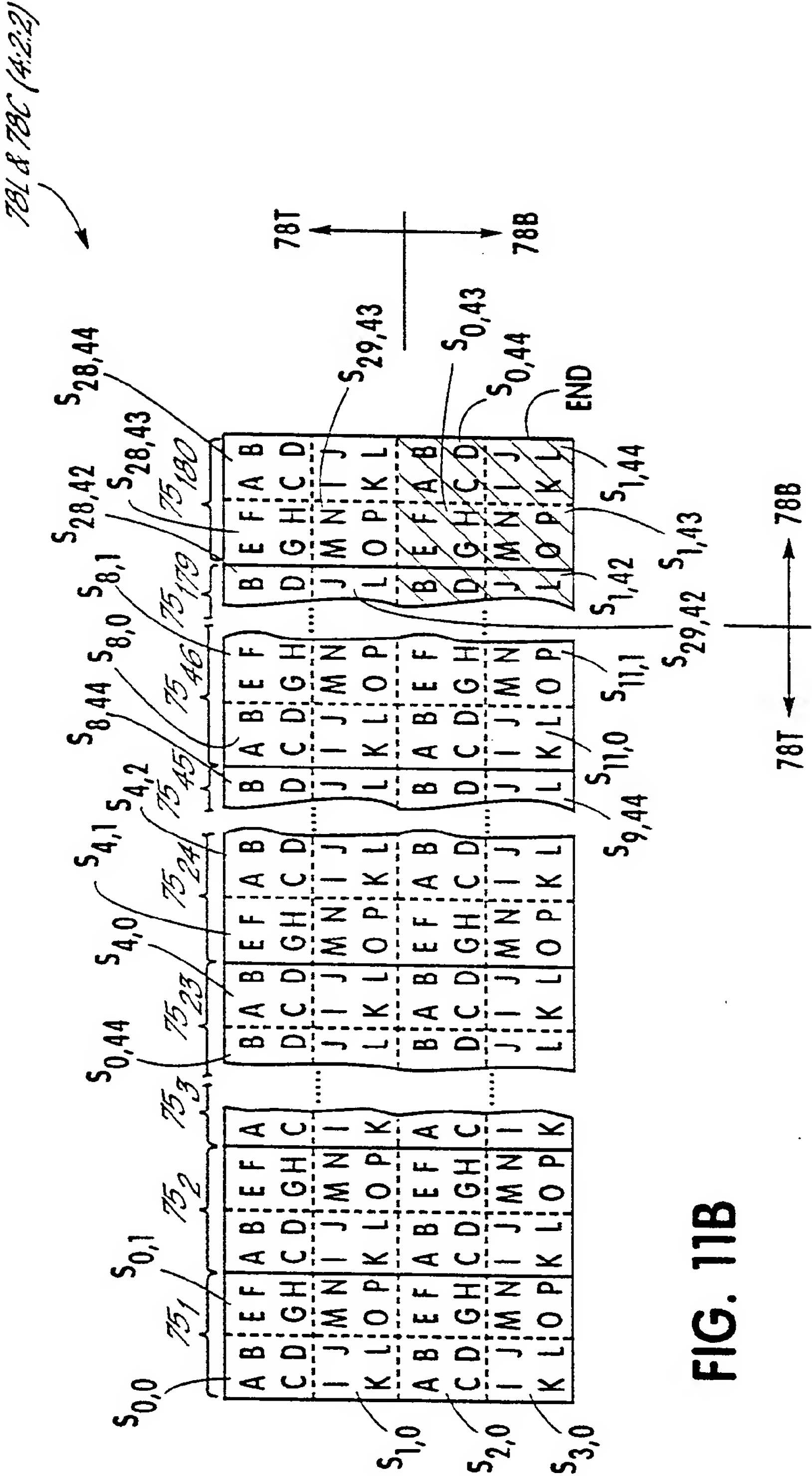


FIG. 11B





**Address calculation****Frame MC****Luma**

$$ax = (ref\_mb\_y[10:5] * hsize) + ref\_mb\_x[10:5], ref\_mb\_x[4]$$

$$ay = ref\_mb\_y[4:1], ref\_mb\_x[3]$$
**Chroma**

$$ax = (ref\_mb\_y[9:5] * hsize + ref\_mb\_x[9:4]), ref\_mb\_x[3]$$

$$ay = ref\_mb\_y[4:1], ref\_mb\_x[2]$$
**Field Mc****Luma**

$$ax = (ref\_mb\_y[10:4] * hsize + ref\_mb\_x[10:5]), ref\_mb\_x[4]$$

$$ay = ref\_mb\_y[3:1], mvfs, ref\_mb\_x[3]$$
**Chroma**

$$ax = (ref\_mb\_y[9:4] * hsize + ref\_mb\_x[9:4]), ref\_mb\_x[3]$$

$$ay = ref\_mb\_y[3:1], mvfs, ref\_mb\_x[2]$$
**Byte Count****Frame Picture**

1 vector luma fetch:  $cy = \sim ref\_mb\_y[3:1]; 111; \& ref\_mb\_y[3]$

1 vector Chroma fetch:  $cy = \sim ref\_mb\_y[3:1]; \& ref\_mb\_y[3:1]$

2 vector luma fetch

$$cy = \sim ref\_mb\_y[3:1]$$

$$cy = ref\_mb\_y[3:1]$$

2 vector Chroma fetch

$$cy = \sim ref\_mb\_y[3:1]$$

2nd fetch  $cy = ref\_mb\_y[2:1]$  required if  $ref\_mb\_y[3]$

**Field Picture**

1 vector luma fetch:  $cy = \sim ref\_mb\_y[3:1]; 111; \& ref\_mb\_y[3]$

1 vector Chroma fetch:  $cy = \sim ref\_mb\_y[3:1]; \& ref\_mb\_y[3:1]$

2 vector luma fetch

$$cy = \sim ref\_mb\_y[3:1]$$

$$cy = ref\_mb\_y[3:1]$$

2 vector Chroma fetch

$$cy = \sim ref\_mb\_y[3:1]$$

2nd fetch  $cy = ref\_mb\_y[2:1]$  required  $ref\_mb\_y[3]$

**FIG. 13**

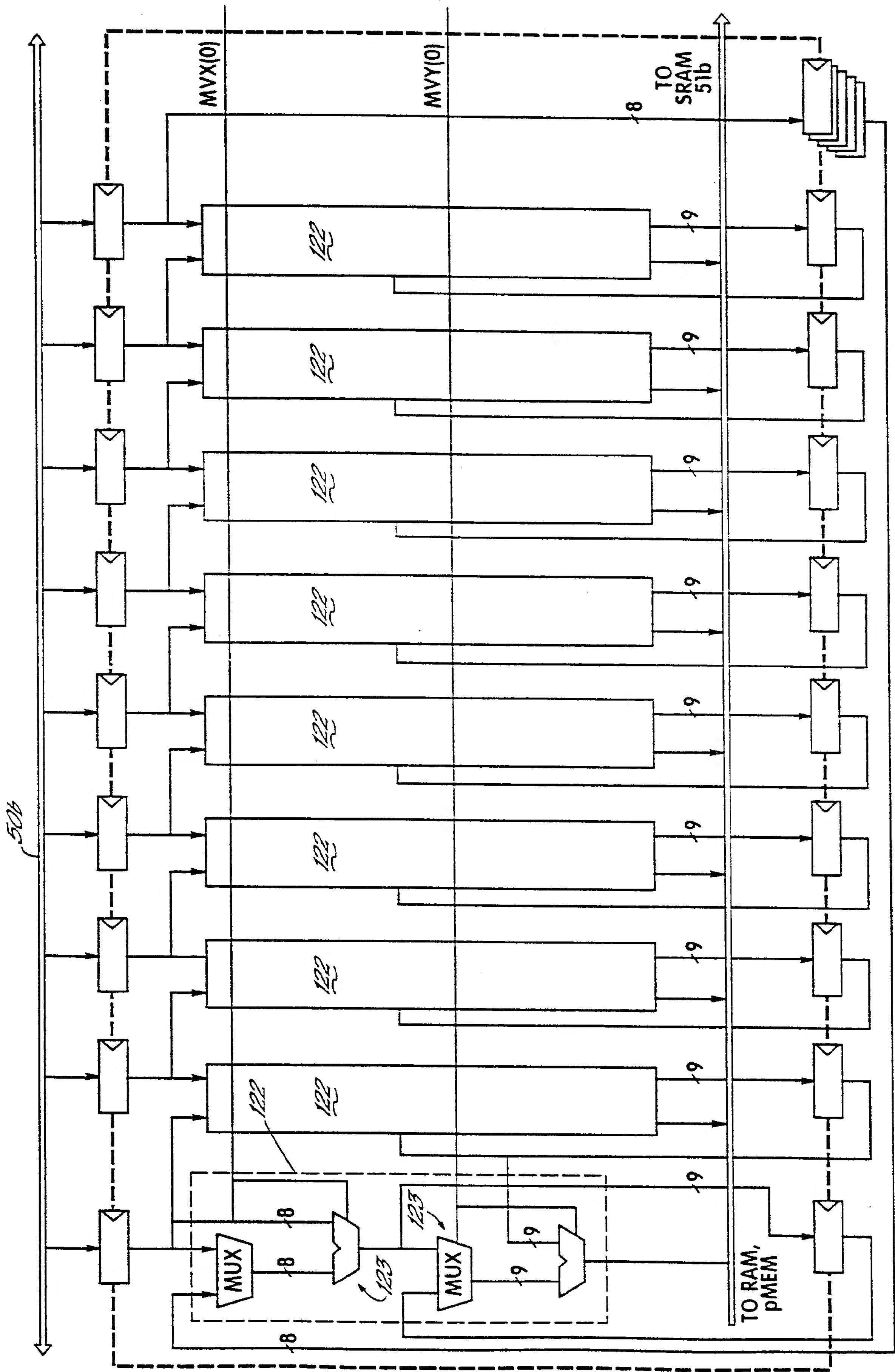


FIG. 15

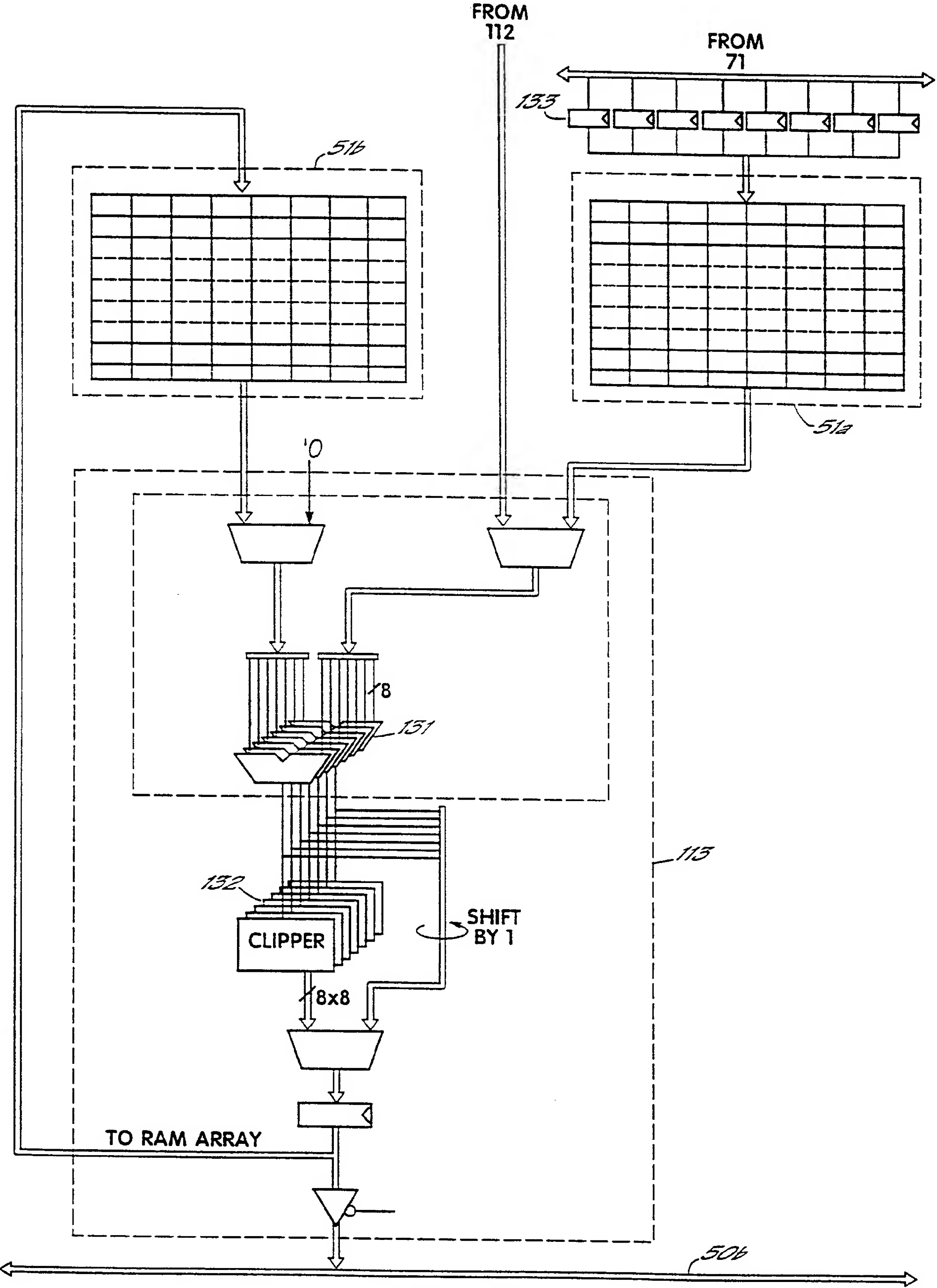
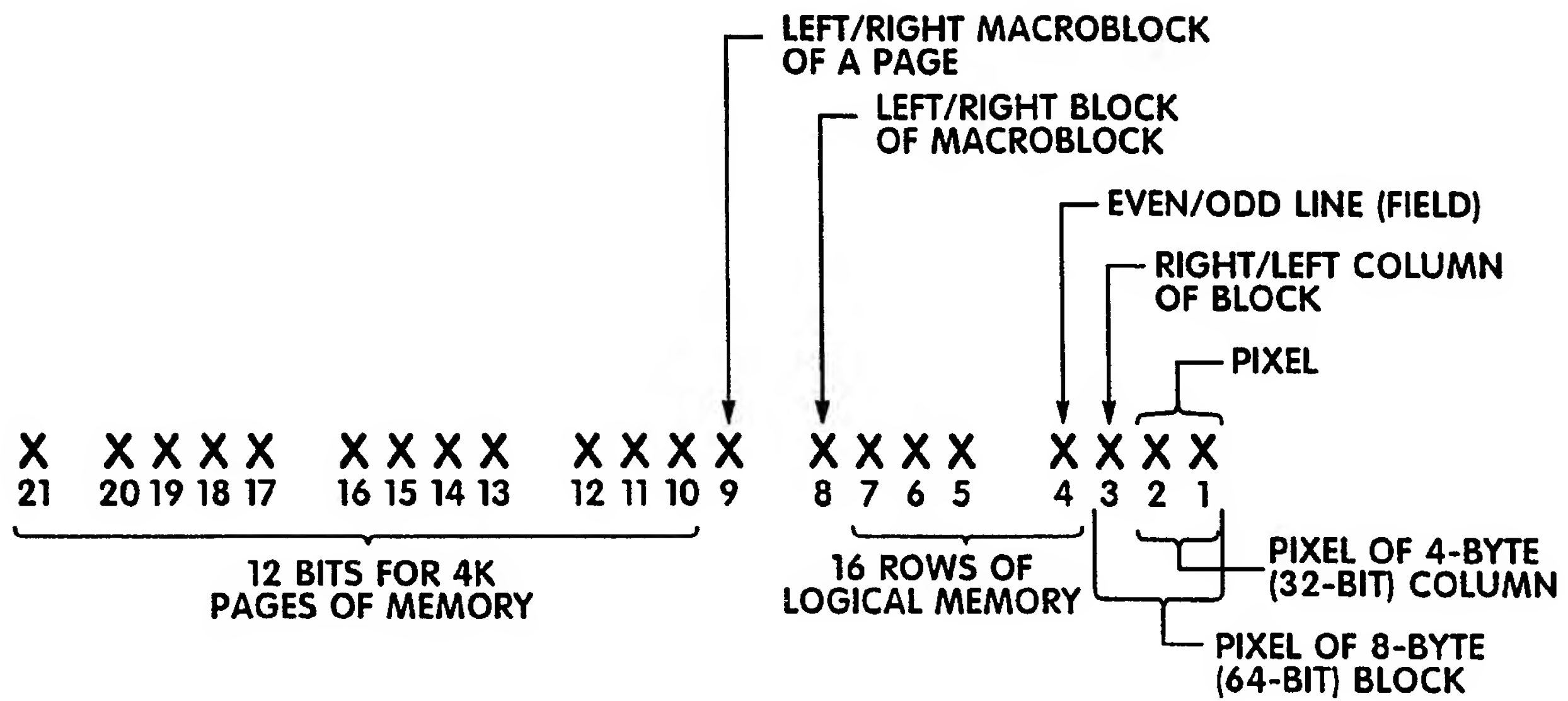


FIG. 15A

19/19

**FIG. 16**



# INTERNATIONAL SEARCH REPORT

International Application No  
PCT/US 98/19097

**A. CLASSIFICATION OF SUBJECT MATTER**  
IPC 6 H04N7/50 H04N7/36

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)  
IPC 6 H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P, X	EP 0 849 953 A (LSI LOGIC CORP) 24 June 1998 see column 11, line 28 - column 13, line 3 see figure 4	1-23
X	WINZKER M ET AL: "ARCHITECTURE AND MEMORY REQUIREMENTS FOR STAND-ALONE AND HIERARCHICAL MPEG2 HDTV-DECODERS WITH SYNCHRONOUS DRAMS" 1995 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS), SEATTLE, APR. 30 - MAY 3, 1995, vol. 1, 30 April 1995, pages 609-612, XP000583293 INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS	1-20
A	see paragraph V	21-23

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance  
"E" earlier document but published on or after the international filing date  
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)  
"O" document referring to an oral disclosure, use, exhibition or other means  
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention  
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone  
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.  
"&" document member of the same patent family

Date of the actual completion of the international search

22 December 1998

Date of mailing of the international search report

30/12/1998

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Berbain, F

# INTERNATIONAL SEARCH REPORT

International Application No  
PCT/US 98/19097

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 697 794 A (GEN INSTRUMENT CORP) 21 February 1996 see column 9, line 51 - column 10, line 45 see figures 2-4 see column 12, line 30 - column 14, line 3	1-18, 21-23
A	see figures 5,11 ---	19,20
A	EP 0 503 956 A (C CUBE MICROSYSTEMS) 16 September 1992 see page 12, line 41 - page 13, line 4 ---	1-23
X	DE 44 41 295 A (SICAN GMBH) 23 May 1996  see page 2, line 58 - page 3, line 1 see page 4, line 61 - page 5, line 5	1-18, 21-23
A	see figure 2 -----	19,20

# INTERNATIONAL SEARCH REPORT

information on patent family members

International Application No

PCT/US 98/19097

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0849953 A	24-06-1998	JP 10294947 A	04-11-1998
EP 0697794 A	21-02-1996	AU 688521 B	12-03-1998
		AU 2729195 A	29-02-1996
		BR 9503640 A	28-05-1996
		CA 2155260 A	16-02-1996
		CN 1121673 A	01-05-1996
		JP 8195960 A	30-07-1996
		NO 953193 A	16-02-1996
		US 5675387 A	07-10-1997
EP 0503956 A	16-09-1992	CA 2062200 A	16-09-1992
		DE 503956 T	28-07-1994
		JP 7131785 A	19-05-1995
		US 5379356 A	03-01-1995
		US 5608888 A	04-03-1997
		US 5649029 A	15-07-1997
		US 5768292 A	16-06-1998
		US 5805488 A	08-09-1998
DE 4441295 A	23-05-1996	EP 0714211 A	29-05-1996